AFRL-PR-WP-TR-2002-2042

THERMAL INTERACTIONS IN ROLLING BEARING DYNAMICS

Pradeep K. Gupta

Pradeep K. Gupta, Inc.
117 Southbury Road
Clifton Park, New York 12065-7714

MARCH 2002

Final Report for 15 April 1996 – 31 December 2001

This is a Small Business Innovation Research (SBIR) Phase II Report.

PROPULSION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7251

NOTICE

Jeffrey H. Brown

JEFFREY H. BROWN, Ph.D., Project Eng.
Mechanical Systems Branch

Robert L. Wright

ROBERT L. WRIGHT, Ph.D., Chief
Mechanical Systems Branch

William E. Koop

WILLIAM E. KOOP
Chief of Technology
Turbine Engine Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
March 2002	Final	04/15/1996 – 12/31/2001

4. TITLE AND SUBTITLE
THERMAL INTERACTIONS IN ROLLING BEARING DYNAMICS

5a. CONTRACT NUMBER
F33615-96-C-2653

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62203F

6. AUTHOR(S)
Pradeep K. Gupta

5d. PROJECT NUMBER
3048

5e. TASK NUMBER
06

5f. WORK UNIT NUMBER
PG

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Pradeep K. Gupta, Inc.
117 Southbury Road
Clifton Park, New York 12065-7714

8. PERFORMING ORGANIZATION REPORT NUMBER
G-120-TR-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Propulsion Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7251

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/PRTM

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-PR-WP-TR-2002-2042

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This is a Small Business Innovation Research (SBIR) Phase II Report. Report contains color.

14. ABSTRACT
Numerical enhancements to the established bearing dynamics computer code, Advanced Dynamics of Rolling Elements (ADORE), have been accomplished for improved computation of bearing heat generation and geometrical variation in bearing element dimensions. Thermal interaction modules have been added to analyze the computed heat generation considering conduction through the element and convection to the circulating fluid or lubricant. Averaging procedures have been implemented to determine the bulk temperatures of the various bearing elements. Well-established methods have been used to compute the associated changes in bearing geometry, which then affect mechanical interaction in the bearing.

15. SUBJECT TERMS
bearings, lubricants, heat generation, software development, SBIR

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor) / 19b. TELEPHONE NUMBER *(Include Area Code)*
Unclassified	Unclassified	Unclassified	SAR	128	Dr. Jeffrey R. Brown (937) 255-7477

HES&S 31-15093-1

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Table of Contents

List of Figures iv
List of Tables vi
Foreword vii
Preface viii

1. Introduction 1

2. Technical Approach to Thermal Modeling 4
2.1 Contact Heat Generation 5
2.2 Heat Partition at Rolling Element to Race Contacts 7
2.3 Conduction Through the Races 8
2.4 Conduction Through the Rolling Elements 9
2.5 Conduction Through the Cage 10
2.6 Convection Through the Coolant 10
2.7 Contact Temperature Modeling 13
2.8 Numerical Implementation 16
2.9 Geometrical Distortions Due to Thermal Effects 20
2.10 Overall Boundary Conditions and Model Implementation 21

3. Other Analytical Models 23
3.1 Generalized Equilibrium Formulations 23
3.2 Fatigue Life Model 27

4. Enhancements to Bearing Dynamics Code ADORE 36
4.1 Conversion to FORTRAN-90 Standard 36
4.2 Procedures for Thermal Interactions 38
4.3 Implementation of Time-Varying Geometry 40
4.4 Other Refinements 40

5. Results 46
5.1 Test Ball Bearing 46
5.2 Moment Equilibrium for Angular Contact Ball Bearings 46
5.3 Thermal Interactions in Ball Bearings 48
5.4 Test Cylindrical Roller Bearing 67
5.5 Thermal Interactions Cylindrical Roller Bearing 68

6. Conclusions 82

7. Recommendations for Future Development 83

8. References 84

Appendix A - Typical ADORE Data Module 86
Appendix B - Typical ADORE Code Segment 90
Appendix C - Ball Bearing Data 102
Appendix D - Cylindrical Roller Bearing Data 109

List of Figures

Figure		Page
1.	Schematic of heat flow from rolling element/race contact to the external system	4
2.	Ball/Race contact schematic	5
3.	Typical slip velocity distribution along the major axis of a ball/race contact	7
4.	Schematic of a cylindrical body	8
5.	Rolling element solid model for heat conduction	9
6.	Contact zone schematic	14
7.	Basic coordinates in the contact zone	15
8.	Grid point schematic for a linear function	17
9.	Comparison of temperature distribution resulting from elliptical and unform heat fluxes	21
10.	Schematic of ball and race relative position	23
11.	Schematic outline of code elements of ADORE	37
12.	Schematic architecture of the thermal module in ADORE	38
13.	Integration of thermal interaction module ADRH with other modules in the bearing dynamics computer code ADORE	39
14.	Ball/Race load distribution at the outer race contact with aligned and misligned condition resulting from moment equilibrium	47
15.	Ball/Race load distribution at the inner race contact with aligned and misligned condition resulting from moment equilibrium	48
16.	Overall powerloss in the test ball bearing with no thermal interactions	50
17.	Time-averaged wear rates in the test ball bearing with no thermal interactions	51
18.	Computed bulk temperatures with no change in bearing geometry	52
19.	Typical ball/race interaction with no change in internal geometry	53
20.	Ball bearing cage mass center velocities in absence of any thermal interactions	54
21.	Ball bearing cage/race contacts in absence of any thermal interactions	55
22.	Ball bearing cage mass center whirl orbits in absence of any thermal interactions	56
23.	Typical ball/cage contacts in absence of any thermal interactions	57
24.	Ball bearing poer loss with time-varying bearing geometry due to thermal effects	59
25.	Time-averaged wear rates for the ball bearing with geometrical distortions due to thermal effects	60
26.	Computed bulk tempeeratues in the ball bearing with thermal effects	61
27.	Typical ball/race interactions with thermal effects	62
28.	Ball bearing cage mass center velocities with changing bearing geometry as a result of thermal effects	63
29.	Ball bearing cage/race contacts with thermal effects	64
30.	Ball bearing cage mass center whirl orbits with thermal effects	65
31.	Typical ball/cage interactions with applied thermal effects	66
32.	Traction-Slip relation for the test roller bearing	68

List of Figures ..continued

Figure Page

33. Roller bearing powerloss with no thermal effects 69
34. Computed bulk temperatures in the roller bearing with no thermal effects 70
35. Typical roller/race interactions with no thermal effects 71
36. Roller bearing cage mass center velocities with no thermal effects 73
37. Cage/race contacts in the roller bearing with no thaermal effects 74
38. Roller bearing cage whirl orbits with no thermal effects 75
39. Roller bearing powerloss with all thermal effects 76
40. Computed roller bearing bulk temperatures with changing bearing geometry as a rsult of thermal effects 77
41. Typical roller/race contacts in the cylindrical roller bearing with changing bearing geometry due to thermal effects 78
42. Change in cage mass center velocities in the roller bearing with changing bearing geometry as a result of thermal effects 79
43. Increased cage/race forces in the roller bearing with thermal effects 80
44. Cage mass center whirl in the roller bearing with thermal effects 81

List of Tables

Table Page

1. Coefficients of Heat Transfer Coefficients for Cylinders in Air 12
2. Physical Properties of Air 13
3. Material Matrix Susceptibility Multiplier 29
4. Steel Processing Multiplier 31
5. Ball Bearing Geometry and Operating Conditions 46
6. Cylindrical Roller Bearing Geometry and Operating Conditions 67

Foreword

The work reported herein was carried out under the Small Business Innovation Research (SBIR) Phase II program. In addition to SBIR, the project development was supported by the Air Force Propulsion Laboratory and NASA Marshall Space Flight Center. The related SBIR Phase I project was carried out under Air Force Contract F33615-95-C-2528. Principal project engineers were Dr. Jeff Brown of the Air Force Propulsion Laboratory and Mr. Howard Gibson of NASA Marshall Space Flight Center.

Preface

Procedures for modeling thermal interactions and implementation of the models in rolling bearing dynamics computer code constitute the principal objectives of the current program. The well established bearing dynamics computer code ADORE (Advanced Dynamics Of Rolling Elements) is taken as the baseline model. Numerical enhancements are carried out for improved computation of bearing heat generation and geometrical variations in the dimensions of bearing elements. With the computed internal heat generation thermal interaction models based on conduction through the bearing elements and convection through the circulating fluid or lubricants are developed. The output from these models is a detailed temperature field throughout the bearing. Certain averaging procedures are then implemented to compute bulk temperature of the various bearing elements. Well established procedures are then implemented to compute change in bearing geometry as a function of temperatures. All mechanical interactions in the bearing are then modified as a function of new geometry. All thermal interactions are free of any transients while the mechanical interactions are controlled by the classical differential equations of motion of the bearing elements. Thus the thermal interactions are implemented in a stepwise averaged manner; for a selected time domain the thermal interactions are averaged before computing the temperature fields and the associated geometrical distortions. Such a stepwise implementation of thermal interactions, although results in a step change in bearing performance parameters, the numerical integration is free of any truncation problems when an explicit algorithm is used, and the integration is inherently convergent. Parametric dynamic simulations based on such modeling procedures show numerically convergent temperature fields as the bearing performance simulations approach steady-state. Thus the model is numerically sound and convergent.

1. Introduction

Rolling element bearings, due to their high load support and stiffness characteristics, serve as the main support bearings in a wide range of practical applications. As the operating conditions become severe in terms of the applied loads and speeds the bearing behavior gets increasingly complex, not just due to the applied external conditions, but the internal interactions and heat generation between the interacting bearing element. It has been well established that frictional interactions and lubrication mechanics are the most critical factors which determine the performance of a rolling bearing. While the friction forces directly contribute to the bearing element motion, the frictional heat generated at the contacts leads to substantial temperature rise of the interacting surfaces, which may appreciably alter the lubricant behavior and therefore, the resulting friction or traction forces. Thus the frictional and thermal phenomena are very closely linked in producing dynamic instabilities in rolling bearings. In addition, the bulk temperature rise of the bearing elements results in a change in the geometrical parameters, such as bearing element size and internal clearance distributions, which again contribute to altered geometrical interaction between the bearing elements, and overall bearing motion. Modeling of thermal interactions in rolling bearings, therefore, has a substantial practical significance.

While the intricate relationship between temperature and traction behavior of liquid lubricants is quite well known, the requirement for higher operating temperatures has recently led to significant interest in the development of high temperature materials and solid lubricants. Since the friction forces are generally higher in a dry contact environment, when compared to liquid lubricated conditions, thermal interactions have increased practical significance for solid lubricated contacts. In fact, the high temperatures, in addition to changing the solid lubricant behavior, often affect bulk properties of the bearing materials. Thus modeling of thermal interaction becomes significant for both bearing behavior and materials development.

The recent development of ceramic bearings has shown that although these advanced materials render acceptable performance at high temperatures, improved modeling of the thermal problem and prediction of temperature distribution is extremely essential in resolving several practical bearing design problems. First, solution to mounting problems resulting from the difference in the thermal coefficient of expansion between ceramics and some of the mating steel parts, require accurate assessment of the temperature fields. Second, and more important, is the fact that the contact temperature, which may be significantly higher than the bulk temperature, greatly affects the chemical phenomena which controls traction behavior of the lubricant. In the case of liquid lubricants, the exponential viscosity-pressure-temperature relations and the shear stress and strain rate behavior have led to the development of acceptable traction models for practical design. In solid lubricants, however, the traction behavior demonstrates certain transitions as a function of temperature. In other words the traction behavior may be subjected to a drastic change as the operating temperature crosses a certain threshold. Such a behavior may trigger instabilities in motion of the bearing elements which may lead to a sudden and very often a catastrophic failure. Precise estimates of internal bearing temperatures and the overall thermal management of the bearing system, therefore, have increased importance in solid lubricated bearings.

While traction behavior of a lubricant depends on the operating temperature, the bearing heat generation is dependent on the traction coefficients, contact loads and operating speeds. Aside from the external applied loads, the internal load distribution in the bearing depends on the operating internal clearance, which in turn is a function of temperature. The variations in internal clearance with temperature are particularly significant in hybrid ceramic bearings due to a significant difference in the thermal coefficient of expansion of the interacting materials. Due to such closely connected frictional and geometrical effects, a realistic modeling of bearing performance requires a solution to the coupled mechanical and thermal interactions between the bearing elements.

In addition to the overall motion of bearing elements and the associated instabilities, the modeling of thermal interactions is very critical in bearings where a thin coating of a certain material is applied to the bearing surfaces to enhance the tribological behavior. Here the contact temperature not only determines the resulting friction or traction coefficients, but also the bond between coating and substrate, since the thermal coefficient of expansion for the two materials may be quite different. To further complicate the problem the contact stresses, which affect the traction forces and bearing heat generation, are also dependent of the coating thickness and the temperature distribution in the composite solid. The problem becomes even more complicated when more than one coating are applied on the interacting surfaces.

Based on the above discussion, practical significance of thermal interactions in rolling bearings may be established on three grounds: (1) mechanical behavior of the materials and structural integrity of fabricated parts, (2) the effect of temperature on traction or friction coefficients at the mating surfaces, and (3) changes in bearing internal geometry due to temperature rise. While the significance of structural integrity is quite obvious, the importance of frictional behavior and changes in bearing internal geometry can be understood in terms of bearing instabilities triggered by internal friction, and the associated thermal effects. A number of investigations [1-5] have shown a strong relationship between bearing failures and instabilities in bearing elements, particularly the cage, triggered by frictional interactions. From a practical standpoint, cage instabilities have several modes [4]: (1) the irregular whirl may cause an erratic variation in bearing torque, (2) the whirl velocities may go through irregular variations to produce bearing squeal, or (3) geometric interactions of the cage may progressively increase with time to produce excessive cage forces and wear. While the first two modes are of interest in precision gyro bearings, the third mode, which often leads to a catastrophic bearing failure, is most important in turbine engine bearings. With a comprehensive model for all thermal and mechanical interactions, parametric modeling studies may reveal an acceptable range of traction parameters, and therefore, a range of thermal environment, over which the motion is stable. Such a study, aside from practical design, provides significant guidance for materials development and selection for a given application.

The thermal effects have an intricate coupling with the geometrical interactions, which affect the applied loads, which in turn control the heat generation and thereby result in a feed back to the thermal problem. Due to such a connection between the thermal, geometrical and mechanical effects, realistic simulation of bearing performance requires a close integration of the thermal models with geometrical interaction and resulting applied loadings. Modeling of all thermal interactions in the bearings and an integration of the thermal models with the models for geometrical interactions is the primary objective of this development project. The existing bearing dynamics

computer code, ADORE [3], is used as a baseline, and the code is enhanced by incorporating the thermal interactions models developed as a part of the current effort.

In addition to incorporation of the thermal interactions, other enhancements to ADORE, include, renewed code structure conforming to the FORTRAN-90 standard, refinement of ball bearing equilibrium module to include moment equilibrium, and enhancements of the fatigue life computations to include the recently developed models for life corrections as a function of the advanced materials behavior and applied operating conditions. Presentation of technical accomplishments during this project are, therefore, divided into several sections. The next section outlines the thermal modeling effort while section 3 is devoted to the moment equilibrium and life enhancement models. ADORE enhancements are reported in section 4, which is followed by some parametric runs with the final version of ADORE.

2. Technical Approach To Thermal Modeling

The main source of heat generation in a rolling bearing is the rolling element to race contact, where for a given set of operating conditions and a prescribed lubricant, local sliding in the contact produces frictional heat. Transfer of this heat to the external environment through the bearing constitutes the basic foundation of the thermal modeling task. Schematically the heat generated in the contact may be partitioned such that part of it goes to the races and part of it goes to the rolling element. The heat going to the races travels via conduction to the external system, while the heat going to the rolling elements is transferred to the circulating lubricant or any other coolant. Thus the lubricant, in addition to providing favorable traction characteristics at the rolling element to race contacts, serves as a coolant. The interactions are shematically described in figure 1 below.

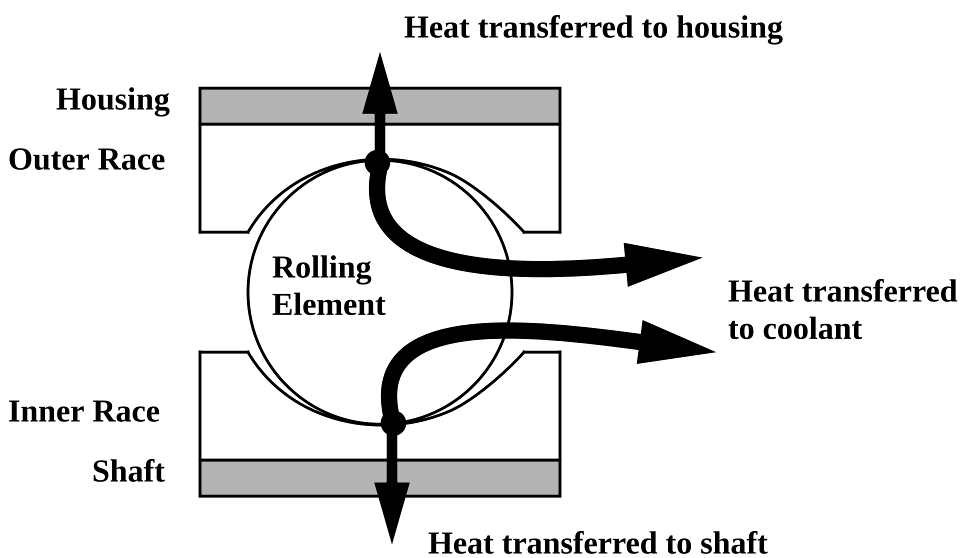

Figure 1. Schematic of heat flow from rolling element/race contact to the external system.

Formulation of thermal interactions, thus consists of the following parts:

- Contact heat generation
- Heat partition between rolling element and race
- Conduction through the races
- Conduction through the rolling element
- Convection to the coolant
- Local temperature rise in the contact

A discussion of the approach used to model each of the above is presented below.

2.1 Contact Heat Generation

Model of computing contact heat generation is quite straightforward. First from the relative position of rolling element and race the normal contact load and size of contact zone are computed. Then for each point in the contact zone the sliding velocity with prescribed rolling element and race velocities is computed. Then using a prescribed traction model, or a traction slip relation, the traction coefficient is computed, which when multiplied by the normal contact load, gives the traction force. Finally, an integration of the product of traction force and sliding velocity over the contact zone gives the total heat generated in the contact.

Figure 2 shows schematically an elliptical contact corresponding to ball/race interaction. Depending on the type of traction model, either variation in traction from point to point in the contact may be permitted, or the variation along the minor axis may be neglected and integration may be performed only along the major axis of the contact ellipse.

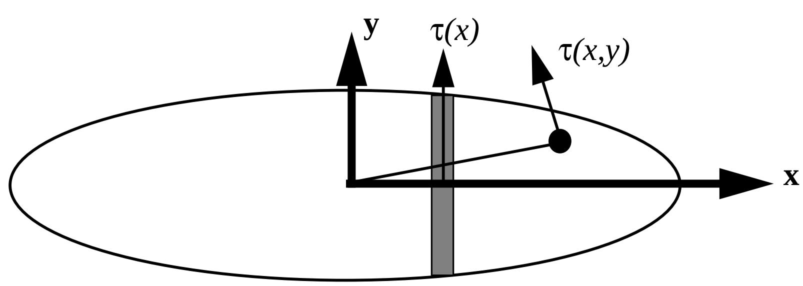

Figure 2. Ball/Race contact schematic.

The shear stress, τ, depends on both normal stress σ and the sliding velocity *u* for a given traction model. The expressions for one and two dimensional variation may be written as:

$$\tau(x) = \sigma(x)\kappa[u(x)] \tag{1}$$

$$\tau(x, y) = \sigma(x, y)\kappa[u(x, y)] \tag{2}$$

Here κ is the traction coefficient, which may be a function of both the sliding velocity and load. For elastohydrodynamic models, the relationship may be quite complex in the sense that variation of lubricant temperatures with pressure and temperature, and variation of lubricant film thickness with load and rolling speed also enter in the model.

In equation (1) corresponding to a one dimensional variation, the sliding velocity is computed only a the point on major axis, while the load is an integrated value over the incremental strip. Also, in this case, only the sliding velocity component along the rolling direction (y) enter in the calculation. In the two dimensional case the direction of slip may be determined by the actual

sliding velocity vector which is defined by the cross product of the angular velocity and the position vector locating the contact point in the contact zone. Thus if **r** and **R** respectively locate the contact point relative to the rolling element and race centers, and ω and Ω are the angular velocity vectors of the rolling element and race, and $\dot{\varphi}$ is the orbital velocity of the rolling element, then the sliding velocity *u* at any point in the contact zone is written as:

$$\mathbf{u} = \omega \times \mathbf{r} - (\Omega - \dot{\varphi}) \times \mathbf{R} \tag{3}$$

Note that the race angular velocity vector normally has only one component as does the rolling element orbital velocity.

Using equation (3) a component of the slip velocity in plane of contact may be determined and then input the traction model to obtain a traction coefficient, which along with the slip velocity may depend on other performance parameters as well.

The total heat generation in the contact, $\bar{q}$, in the contact is now simply an integration of equation (1) or equation (2) depending on one or two dimensional variation.

$$\bar{q} = \int \langle \int \sigma(x, y) dy \rangle \kappa[u(x)] dx \tag{4}$$

or

$$\bar{q} = \int\int \sigma(x, y) \kappa[u(x, y)](dx) dy \tag{5}$$

Implementation of the above equation in the bearing dynamics codes is done numerically since both the traction model and the slip distribution may be rather complicated functions of the position coordinates. Equation (5) is implemented by using numerical quadrature similar to the classical Chebyshev integration procedures [6], while the conventional Gaussian quadrature formulae [7] may be used in implementing equation (4). However, when the slip velocities are computed along the major axis of the contact ellipse, there may be discontinuities at the points of pure rolling, as shown in figure 3. Under such conditions, direct application of the Gaussian quadrature formula over the entire contact length may provide accurate results. Thus the computation is divided into two parts. First the roots of the nonlinear slip equation (3) are computed in the range of contact limits. The roots define the pure rolling points, as shown in figure 3. The contact zone is now divided into subzones by using the pure rolling points as the terminal points for the zone; the slip distribution shown in figure 3 has three zones as shown. The Gaussian quadrature formula is now applied in each zone and the total integral value is obtained by summation. Such an implementation is found to provide fairly accurate results.

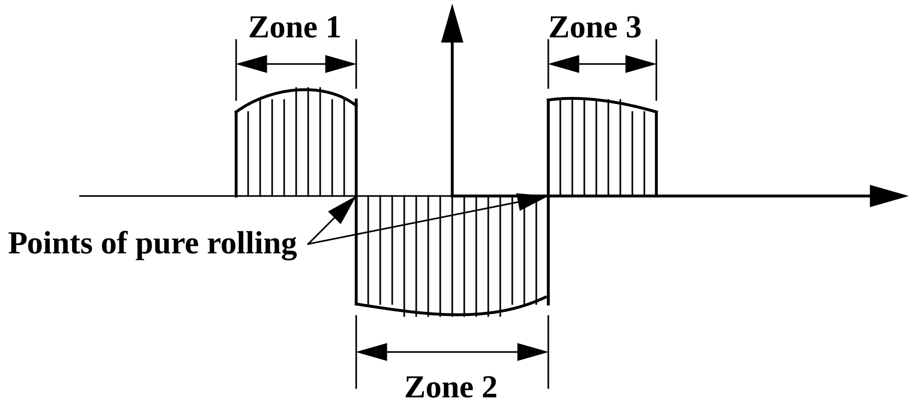

Figure 3. Typical slip velocity distribution along the major axis of a ball/race contact.

2.2 Heat Partition at Rolling Element to Race Contact

In concentrated rolling/sliding contacts, the rolling speed normally pumps the lubricant in the contact and a lubricating film is formed. When sliding is imposed, the film is sheared to produce a traction force and the frictional energy is dissipated as heat. A part of this heat may be carried, via convection, by the circulating lubricant, and the remainder may be transferred from the film to the mating surfaces via conduction. For most machine elements, Trachman and Cheng [8] have shown that the convective part is quite small and it can be reasonably neglected. Thus if the total heat flux at the contact is q and it is partitioned into two parts, q_1 and q_2, which are transferred to the rolling element and race:

$$q_1 = \gamma q \tag{6}$$

and

$$q_2 = (1 - \gamma) q \tag{7}$$

If the two surfaces are separated by a film with thickness, h, and they are at temperatures T_1 and T_2, then there should be an additional conduction term:

$$q_1 = \frac{k_f}{\mathrm{h}}(T_2 - T_1) + \gamma q \tag{8}$$

and

$$q_2 = \frac{k_f}{\mathrm{h}}(T_1 - T_2) + (1 - \gamma) q \tag{9}$$

Here k_f is the thermal conductivity of the film and γ is the heat partition fraction. The value of this heat partition fraction will depend on the thermal properties of the two interacting bodies. The actual value will be an outcome of the contact temperature computation discussed later in section 2.6.

2.3 Conduction Through the Races

Conduction through the races, housing and a hollow shaft, can be all be modeled by classical text book approach [9-11]. Consider a cylindrical element of length L, outer and inner radii of r_1 and r_2 at temperatures T_1 and T_2 respectively, as shown in figure 4.

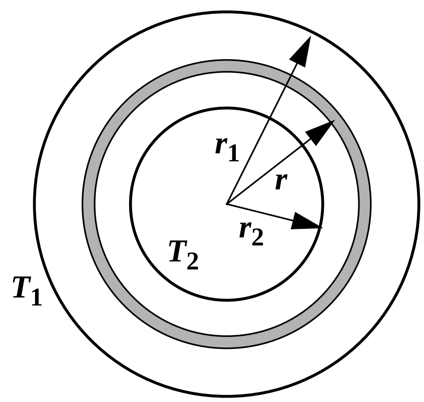

Figure 4. Schematic of a cylindrical body.

The heat flow, q, through an elementary section, as shown in figure 4, is simply

$$q = -kA\frac{dT}{dr} \tag{10}$$

Where *k* is the thermal conductivity and the area of the incremental section, $A = 2\pi rL$, L being length of the cylindrical element.

The temperature distribution along the radius is obtained by simple integration of equation (10):

$$T = T_2 - \frac{q}{2\pi Lk}\ln\frac{r}{r_2} \tag{11}$$

The bulk temperature of the cylindrical element may be approximated by an average temperature, T_{av}, which is obtained by integrating equation (11) with respect to the radius *r*.

$$T_{av} = T_2 - \frac{q}{2\pi Lk}\left[\frac{r_1 \ln\frac{r_1}{r_2}}{r_1 - r_2} - 1\right] \tag{12}$$

The above formulation may also be applied to an angular sector of the cylinder, with an arc length, θ, by simply replacing the term, 2π by the angle, θ. This becomes useful when conduction through the race is carried out at each rolling element location, where the race surface temperature may be different due to variation in contact load, slip velocity and traction characteristics.

2.4 Conduction Through the Rolling Element

Temperature distribution in the rolling element can be very complicated. With the prescribed heat input at the race contacts and certain boundary conditions, the two or three dimensional partial differential equation may certainly be solved with either finite difference or finite element techniques. However, implementation of such a procedure in the bearing dynamics code will be practically impossible due to constraints on the required computing effort for solving the differential equation of motion of the bearing elements. Thus for the present investigation the rolling is modeled by a cylindrical core with a length equal to the rolling element diameter, and cross sectional area determined such that the total volume of this core is equal to the rolling element. Schematic of the model is shown in figure 5. Thus the equivalent cross sectional will be the rolling element volume divided by its diameter:

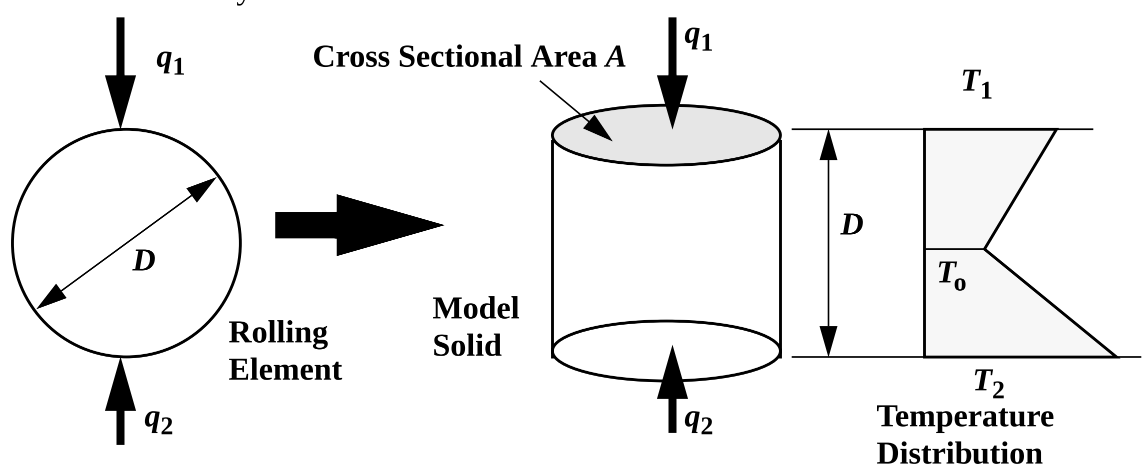

Figure 5. Rolling element solid model for heat conduction.

$$A = \frac{1}{6}\pi D^2 \text{ for ball bearings} \tag{13}$$

and

$$A = \frac{1}{4}\pi DL \text{ for roller bearings.} \tag{14}$$

The conduction equations are simply

$$q_1 = k(T_1 - T_0)\frac{A}{D} \tag{15}$$

and

$$q_2 = k(T_2 - T_0)\frac{A}{D} \tag{16}$$

Here T_1 and T_2 are the contact temperatures and T_o is the mid point temperature. Normally the heat inputs at the outer and inner race contacts (q_1, q_2) are given. So with the temperature known at either the inner or outer race contact, the other temperature and the temperature at mid point can be computed by simultaneous solution of equations (15) and (16).

The average temperature is simply computed by the shaded area under the linear temperature distribution shown in figure 5.

$$T_{av} = \frac{T_o}{2} + \frac{T_1 + T_2}{4} \tag{17}$$

2.5 Conduction Through the Cage

Model for conduction through the cage is identical to that discusses above for the races. The cage is modeled as a cylindrical element and the heat is assumed to conduct radially away from the surface being guided on the race. Heat generated at the cage/race land and all pockets constitutes the total heat flux. Base temperature of the cage is set at the average ambient temperature in the vicinity of the rolling elements. More rigorous temperature distribution in the cage is excluded from the present model. Such an analysis may be best carried out by a finite element model where the heat generation values, as computed in ADORE, may be input to model detailed temperature distribution in the vicinity of all cage contacts.

2.6 Convection Through the Coolant

Convective heat transfer is generally expressed in terms of three dimensionless groups [10, pp 265-268], [11, pp 410-415], [12, pp 200-202]

Reynolds Number: $$Re = \frac{V_\infty \rho_\infty D}{\mu_f} \tag{18}$$

Nusselt Number: $$Nu = \frac{hD}{k_f} \tag{19}$$

Prandtl Number: $$\frac{c_p\mu_f}{k_f} \tag{20}$$

Here V_∞, ρ_∞, μ_f, k_f, c_p, D and h are respectively the characteristic fluid velocity, density, viscosity, thermal conductivity, specific heat, characteristic length of the solid and the heat transfer coefficient. Normally the relationship for heat transfer coefficients are expressed in terms of empirical algebraic equations containing the above three group.

Pertinent to rolling bearings the solids of interest are spheres and cylinders, and the fluids of interest are air and liquids. Air may be used for solid lubricated bearings while the formulae for liquids are used for oil lubricated bearings. The formulae used in the current model are presented below.

2.6.1 Spheres in Air

The common equations for spheres in air are:

$$Nu = 0.37Re^{0.60}, \ 25 \le Re \le 100,000 \tag{21}$$

and

$$Nu = RePr\left(\frac{2.2}{Re} + \frac{0.480}{Re^{0.50}}\right), \ 1 \le Re \le 25 \tag{22}$$

2.6.2 Spheres in Liquids

For spheres in liquids the following relation may be used for all Reynolds numbers:

$$Nu = Pr^{0.30}[0.97 + 0.68Re^{0.50}] \tag{23}$$

In the limiting case, with $Re < 1$, and $Pr = 1$, the Nusselt number approaches a value of 2.

2.6.3 Cylinders in Air

For cylinders in air, the polynomial relationship between the Nusselt and Reynold numbers has a larger variation as a function of the Reynolds number. The suggested expression is:

$$Nu = CRe^n \tag{24}$$

Where constants C and n are tabulated below, in Table 1, as a function of the Reynolds number.

Table 1: Coefficients of Heat Transfer Coefficients for Cylinders in Air

Re	*C*	*n*
0.40 - 4	0.891	0.330
4 - 40	0.821	0.385
40 - 4,000	0.615	0.466
4,000 - 40,000	0.174	0.618
40,000 - 400,000	0.0239	0.805

2.6.4 Cylinders in Liquids

Similar to spheres in liquids, one expression may be used for all Reynolds numbers for cylinders in liquids:

$$Nu = [0.35 + 0.56Re^{0.50}]Pr^{0.31} \tag{25}$$

An equation similar to that described above for air has also been suggested for cylinders in liquids:

$$Nu = 1.1CRe^{n}Pr^{0.31} \tag{26}$$

Here the constants C and n are same as those listed in table 1 for air. However, in the present investigation equation (25) is used for cylinders in liquids.

2.6.5 Properties of Air

Required properties of air for applying the above models for convective heat transfer in air are documented below.

Density at atmospheric pressure is expressed as a function of temperature *T* (^oK) as [13]:

$$\rho = \frac{353}{T}\ \text{kg/m}^3 \tag{27}$$

Other required properties are extracted from ref [10, pp 483], converted to current SI units, and are tabulated below in Table 2:

Table 2: Physical Properties of Air

Temperature $({}^oK)$	Specific Heat $\left(\frac{Nm}{kg^oK}\right)$	Viscosity $\left(\frac{Ns}{m^2}\right)$	Thermal Conductivity $\left(\frac{N}{s^oK}\right)$
88.88	1.0019409E+03	6.3274590E-06	8.2733628E-03
144.44	1.0019409E+03	1.0008138E-05	1.3206225E-02
200	1.0019409E+03	1.3316613E-05	1.8000622E-02
227.77	1.0023596E+03	1.4888138E-05	2.0423783E-02
273.33	1.0031970E+03	1.7245428E-05	2.4231607E-02
311.11	1.0052905E+03	1.9023733E-05	2.7174014E-02
366.66	1.0098962E+03	2.1546445E-05	3.1328006E-02
422.22	1.0136644E+03	2.3903733E-05	3.5308913E-02
477.77	1.0241318E+03	2.6054243E-05	3.8943654E-02
533.33	1.0358553E+03	2.8080685E-05	4.2578392E-02
588.88	1.0484163E+03	3.0024413E-05	4.6040051E-02
644.44	1.0613959E+03	3.1844074E-05	4.9328627E-02
700	1.0743754E+03	3.3622379E-05	5.2444120E-02
755.55	1.0877737E+03	3.5317973E-05	5.5386531E-02
811.11	1.1011721E+03	3.6889498E-05	5.8328938E-02
922.22	1.1250377E+03	3.9991194E-05	6.3867588E-02

2.7 Contact Temperature Modeling

A model for estimate of contact temperature rise for a given heat generation was developed in the first phase of this project [14]. The model is based on the classical flash temperature theory of moving heat sources, as documented by Jaeger [9]. The modeling procedure is reviewed here for completeness.

As shown in figure 6, let the generalized elliptical contact zone between the rolling element and race be divided into incremental strips. The temperature distribution, *T(y)*, along the rolling direction (y), on the incremental strip is given by the equation

$$T(y) = T_0 + \frac{1}{\sqrt{\pi\rho c k U}} \int_{-b}^{y} \frac{q(y)}{\sqrt{y-\xi}} d\xi \tag{28}$$

where T_o is a reference temperature and ρ, c, k, U are respectively the density, heat capacity, thermal conductivity, and rolling velocity. The heat flux *q(y)* for a given pressure distribution *p(y)*, traction coefficient κ, and sliding velocity u_s, is simply written as

$$q(y) = \kappa u_s p(y) \tag{29}$$

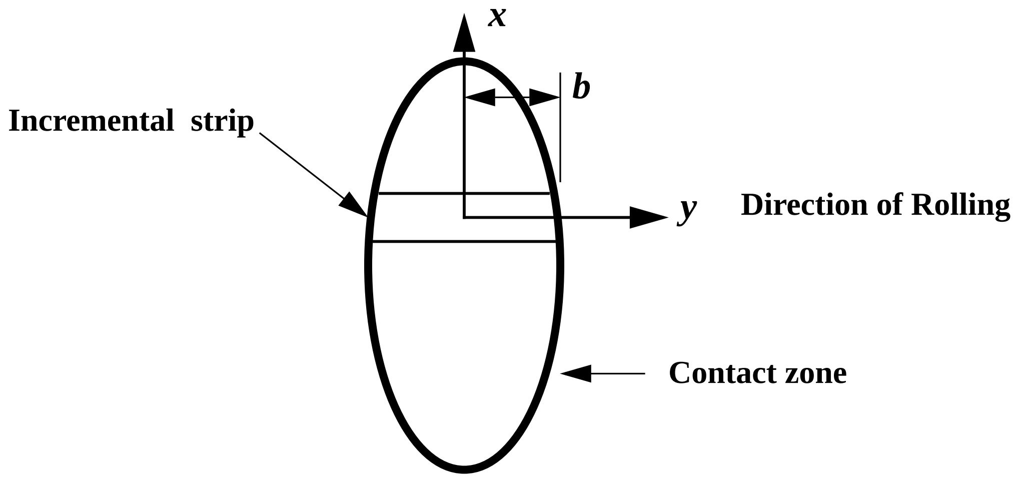

Figure 6. Contact zone schematic.

In accordance to the basic equations, (8) and (9) presented earlier, if the two surfaces are separated by a lubricant film of thickness, h, and the heat generated in the contacts is partitioned into the two surfaces by fractions γ and $(1-\gamma)$, then the heat fluxes going into surfaces 1 and 2 at any coordinate position, *y*, as shown schematically in figure 7 are given by

$$q_1(y) = \frac{k_f}{\mathrm{h}}(T_2 - T_1) + \gamma q(y) \tag{30}$$

$$q_2(y) = \frac{k_f}{\mathrm{h}}(T_1 - T_2) + (1-\gamma) q(y) \tag{31}$$

where k_f is the thermal conductivity of the fluid, or lubricant. The corresponding temperature distributions on the two surfaces, denoted by subscripts 1 and 2, may be written as

$$T_1(y) = T_0 + \frac{1}{\sqrt{\pi \rho_1 c_1 k_1 U_1}} \int_{-b}^{y} \frac{q_1(y)}{\sqrt{y-\xi}} d\xi \tag{32}$$

$$T_2(y) = T_0 + \frac{1}{\sqrt{\pi\rho_2 c_2 k_2 U_2}} \int_{-b}^{y} \frac{q_2(y)}{\sqrt{y-\xi}} d\xi \tag{33}$$

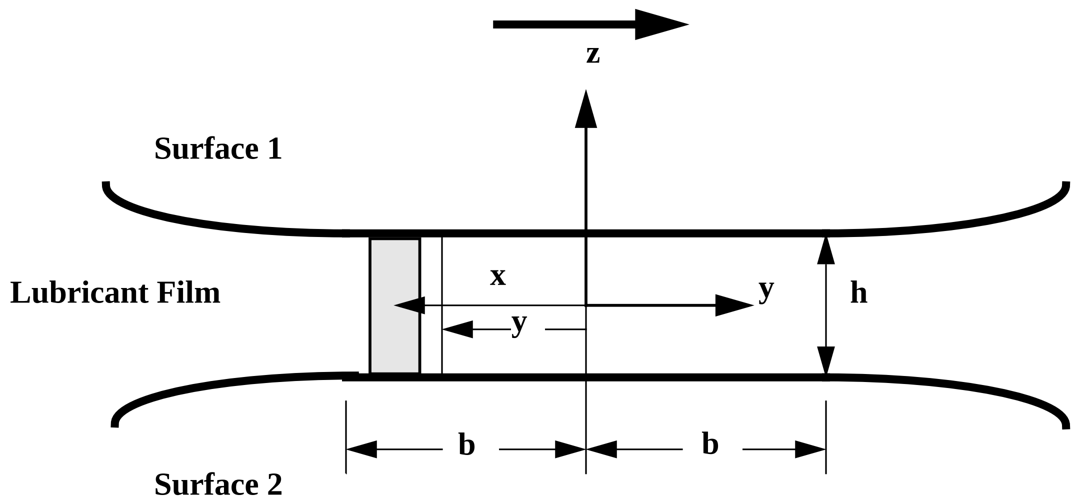

Figure 7. Basic coordinates in the contact zone.

When the two surfaces are assumed to be equal temperatures, the conduction term in equations (30) and (31) vanishes, and the heat partition fraction reduces to

$$\alpha = \frac{1}{\sqrt{\pi\rho_1 c_1 k_1 U_1} + \sqrt{\pi\rho_2 c_2 k_2 U_2}} \tag{34}$$

The expression for temperature rise, $\Delta T(y) = T(y) - T_o$, in the contact along the rolling direction thereby reduces to

$$\Delta T(y) = \frac{1}{\sqrt{\pi\rho_1 c_1 k_1 U_1} + \sqrt{\pi\rho_2 c_2 k_2 U_2}} \int_{-b}^{y} \frac{q(y)}{\sqrt{y-\xi}} d\xi \tag{35}$$

The above equation gives the local temperature rise over an incremental strip in the contact along the rolling direction. For computation of average temperature this equation may be integrated over the contact area

$$T_{ave} - T_o = \frac{1}{\pi ab}\int_{-a}^{a}\int_{-b}^{b}\Delta T(y)dy\ dx \tag{36}$$

where *a* is the contact half width in the direction normal to rolling.

2.8 Numerical Implementation

Although the implementation of the above equations, when the two surfaces are at equal temperatures, is rather straightforward, a generalized numerical procedure is presented below and the equal temperature case is discussed later as a special case. Thus a close examination of the general equations (28) to (31) reveals a recursive relationship for the computation of the required temperature distribution. For the purpose of developing such a relationship and the associated numerical procedure, it is convenient to introduce the following dimensionless variations

$$T' = \frac{T}{T_o}.\ \tau' = \frac{\tau}{\tau_o},\ y' = \frac{y}{b},\ \xi' = \frac{\xi}{b},\ u' = \frac{u}{U} = \frac{|U_2 - U_1|}{U},\ q' = \frac{q}{\tau_o U},\ A = \frac{k_f}{\mathrm{h}}\frac{\sqrt{b}}{\sqrt{\pi\rho ckU}},$$

and $B = \dfrac{\tau_o U\sqrt{b}}{T_o\sqrt{\pi\rho ckU}}$

In terms of the above variables, equations (28) to (31) may be combined to yield:

$$T_1(y) = 1 + A_1\int_{-1}^{y}\left(\{T_2(\xi) - T_1(\xi)\}\frac{d\xi}{\sqrt{y-\xi}}\right) + B_1\int_{-1}^{y}\frac{\gamma q(\xi)}{\sqrt{y-\xi}}d\xi \tag{37}$$

and

$$T_2(y) = 1 + A_2\int_{-1}^{y}\left(\{T_1(\xi) - T_2(\xi)\}\frac{d\xi}{\sqrt{y-\xi}}\right) + B_2\int_{-1}^{y}\frac{(1-\gamma)q(\xi)}{\sqrt{y-\xi}}d\xi \tag{38}$$

where the subscripts $_1$ and $_2$ on the constants *A* and *B* represent the corresponding properties of the two surfaces. Also, the superscript, prime (‘), has been dropped for brevity.

Note that although the required temperatures T1 and T2 appear on both side of the equations, the integrating limits are such that no values of temperatures for coordinates $\xi > y$ are required. Therefore, a recurrence type of relation can be developed by expressing these equations in discrete form. In order to develop the numerical form of such a relation, it will be convenient to develop certain quadrature, which can be readily used to evaluate the integral expressions. Consider a function $f(\xi)$ to be linear between the grid points *j-1* and *j* corresponding to coordinates ξ_{j-1} and ξ_j, as shown shematically in figure 8. In other words:

$$f(\xi) = f_{j-1} + \frac{f_j - f_{j-1}}{\xi_j - \xi_{j-1}}(\xi - \xi_{j-1}) \tag{39}$$

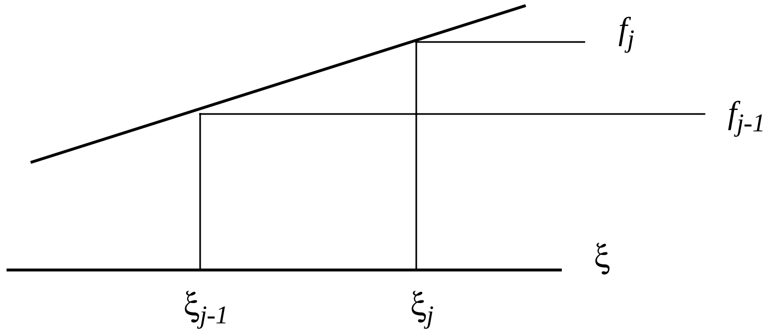

Figure 8. Grid point schematic for a linear function.

Using the above linearized form of a function, two quadrature, *I* and *J* may be expressed to define the integrals:

$$I_j^i = \int_{\xi_{j-1}}^{\xi_j} \frac{d\xi}{\sqrt{\xi_i - \xi}} = 2[\sqrt{\xi_i - \xi_{j-1}} - \sqrt{\xi_i - \xi_j}] \tag{40}$$

and

$$J_j^i = \int_{\xi_{j-1}}^{\xi_j} \frac{\xi d\xi}{\sqrt{\xi_i - \xi}} = \frac{2}{3}[(2\xi_i + \xi_{j-1})\sqrt{\xi_i - \xi_{j-1}} - (2\xi_i + \xi_j)\sqrt{\xi_i - \xi_j}] \tag{41}$$

In addition, an integral of the function $f(\xi)$ between the two grid points may be defined by the quadrature:

$$S(f)_{ij} = \int_{\xi_{j-1}}^{\xi_j} \frac{f(\xi)d\xi}{\sqrt{\xi_i - \xi}} + \left(f_{j-1} + \frac{f_j - f_{j-1}}{\xi_j - \xi_{j-1}}\xi_{j-1}\right)I_j^i + \left(\frac{f_j - f_{j-1}}{\xi_j - \xi_{j-1}}\right)J_j^i \tag{42}$$

Using the above quadrature the integral terms in equations (37) and (38) may be represented in the following discrete form:

$$\begin{aligned}\int_{-1}^{\xi_i} \frac{f(\xi)d\xi}{\sqrt{\xi_i - \xi}} &= \sum_{j=2}^{i-1} S(f)_{ij} + \left(f_{i-1} + \frac{f_i - f_{i-1}}{\xi_i - \xi_{i-1}}\xi_{i-1}\right)I_i^i + \left(\frac{f_i - f_{i-1}}{\xi_i - \xi_{i-1}}\right)J_i^i \\ &= \sum_{j=2}^{i-1} S(f)_{ij} + f_{i-1}\left(\frac{I_i^i\xi_i - J_i^i}{\xi_i - \xi_{i-1}}\right) + f_i\left(\frac{J_i^i - I_i^i\xi_{i-1}}{\xi_i - \xi_{i-1}}\right)\end{aligned} \tag{43}$$

Now substitution of the above in equations (27) and (38) gives the discrete expressions for temperatures at the various grid points:

$$T_{1i} = 1 + A_1 \sum_{j=2}^{i-1} [S(T_2)_{ij} - S(T_1)_{ij}] + B_1 \gamma \sum_{j=2}^{i} S(q)_{ij} \tag{44}$$
$$+ T_{2i-1} A_1 \left(\frac{I_i^i \xi_i - J_i^i}{\xi_i - \xi_{i-1}} \right) + T_{2i} A_1 \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right)$$
$$+ T_{1i-1} A_1 \left(\frac{I_i^i \xi_i - J_i^i}{\xi_i - \xi_{i-1}} \right) + T_{1i} A_1 \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right)$$

and

$$T_{2i} = 1 + A_2 \sum_{j=2}^{i-1} [S(T_1)_{ij} - S(T_2)_{ij}] + B_2 (1 - \gamma) \sum_{j=2}^{i} S(q)_{ij} \tag{45}$$
$$+ T_{1i-1} A_2 \left(\frac{I_i^i \xi_i - J_i^i}{\xi_i - \xi_{i-1}} \right) + T_{1i} A_2 \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right)$$
$$+ T_{2i-1} A_2 \left(\frac{I_i^i \xi_i - J_i^i}{\xi_i - \xi_{i-1}} \right) + T_{2i} A_2 \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right)$$

For the purpose of implementing the above equations in a computer subprogram, the above equations may be written in the following matrix form:

$$[C_{kl}]\{T_l\} = \{R_k\} \tag{46}$$

where the various components are written as

$$C_{11} = 1 + \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right) A_1 \qquad C_{12} = -\left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right) A_1$$

$$C_{21} = -\left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right) A_2 \qquad C_{22} = 1 + \left(\frac{J_i^i - I_i^i \xi_{i-1}}{\xi_i - \xi_{i-1}} \right) A_2$$

$$R_1 = 1 + A_1 \left(\sum_{j=2}^{i-1} [S(T_2)_{ij} - S(T_1)_{ij}] \right) + B_1 \gamma \sum_{j=2}^{i} S(q)_{ij}$$
$$+ \left(\frac{I_i^i \xi_i - J_i^i}{\xi_i - \xi_{i-1}} \right) (T_{2i-1} - T_{1i-1}) A_1$$

$$R_2 = 1 + A_2\left(\sum_{j=2}^{i-1}[S(T_2)_{ij} - S(T_1)_{ij}]\right) + B_2(1-\gamma)\sum_{j=2}^{i} S(q)_{ij}$$
$$+ \left(\frac{I_i^i\xi_i - J_i^i}{\xi_i - \xi_{i-1}}\right)(T_{1i-1} - T_{2i-1})A_1$$

Solution to the above equations for the temperatures T_1 and T_2 is now straightforward, when the shear is known over the contact, and it is independent of temperature. In the event the shear stress depends on the temperature, then a certain iterative procedure shall be necessary to determine compatible shear stress and temperatures. Alternatively, if no temperature dependence of shear stress is assumed between two consecutive nodes, *i-1* and *i*, then q_i may simply be calculated at temperature T_{i-1} and the iterative procedure may be conveniently eliminated. As the grid size becomes smaller such a simplification may become increasingly reasonable.

A special case of the above formulation is when the temperatures of the two interacting surfaces are assumed to be equal. Such a case is certainly applicable to solid lubricant bearings. For liquid lubricated bearings, particularly when the lubricant film thickness is significant, such a simplification is indeed an assumption. In view of the fact, that for most of the experimental traction data and subsequent regression analysis, the contact temperatures are assumed to be the same, since no reliable experimental values are available, the assumption of equal surface temperatures may not be unreasonable for liquid lubricated bearings as well. Thus such an assumption in made in the current development. A result of such an assumption is elimination of the conduction terms in the above formulation, which is indeed a significant numerical simplification.The base equations for temperature distribution are now written as:

$$T_1(y) = B_1\gamma\int_{-1}^{y}\frac{q(\xi)}{\sqrt{y-\xi}}d\xi \tag{47}$$

and

$$T_2(y) = B_2(1-\gamma)\int_{-1}^{y}\frac{q(\xi)}{\sqrt{y-\xi}}d\xi \tag{48}$$

Since $T_1 = T_2$, the heat partition fraction is simply expressed in terms of the material properties

$$\gamma = \frac{B_2}{B_1 + B_2} \tag{49}$$

Thus the temperature equation may now be written as

$$T(y) = 1 + \frac{B_1 B_2}{B_1 + B_2} \int_{-1}^{y} \frac{q(\xi)}{\sqrt{y - \xi}} d\xi \tag{50}$$

and in discrete form

$$T_i = 1 + \frac{B_1 B_2}{B_1 + B_2} \sum_{j = 2}^{i} S(q)_{ij} \tag{51}$$

Arguments with regard to temperature dependence of shear stress apply as discussed in the general case. When the shear stress does not depend on temperature, the integration is straighforward. Now if we define a dimensionless temperature rise at any point *i*, in the contact, as

$$\Delta\tilde{T}_i = \frac{(T_i - 1)(B_1 + B_2)}{B_1 B_2} \tag{52}$$

then equation (51) may be expressed in the form of temperature rise as:

$$\Delta\tilde{T}_i = \sum_{j = 2}^{i} S(q)_{ij} \tag{53}$$

To elaborate on the results obtained with the above equations, consider an elliptical contact with a Hertzian pressure distribution. Also, assume a constant coefficient of friction and sliding velocity. Under such conditions, the heat flux will also be elliptical. The results obtained by implementing equation (53) under such conditions are plotted below in figure 9, along with the results obtained with a uniform heat flux.

2.9 Geometrical Distortions Due to Thermal Effects

As the heat travels through the bearing elements the overall temperatures of the various elements will change which will result in change in internal clearances, and, therefore, the applied load. Models used to compute such geometrical distortions are based on text book formulae for cylindrical and spherical elements as documented in the literature. These formulae are very well understood and they are, therefore, omitted here for brevity.

When changes in bearing geometry are applied, a change in bearing internal clearance will lead to a step change in performance parameters, such as contact load, when the races are held in a fixed position. If the applied loads are required to remain constant then it will be necessary to impose an equilibrium constraint and move the races accordingly. Note also, that there are no thermal transients in the current model. Thus changes in bearing geometry will be instantaneous when applied. Considering such limitations, the local heat generations are first averaged over a certain time domain before computing the temperature field. Then the geometrical changes are applied in a step wise fashion at selected time steps. Both the averaging times and the times for implementation of geometrical changes are inputs to the numerical procedure. Such a step wise

procedure is justified in absence of any thermal transients. Also, when explicit procedures are used to integrate the equations of motion, such an implementation will be free of any truncation problems.

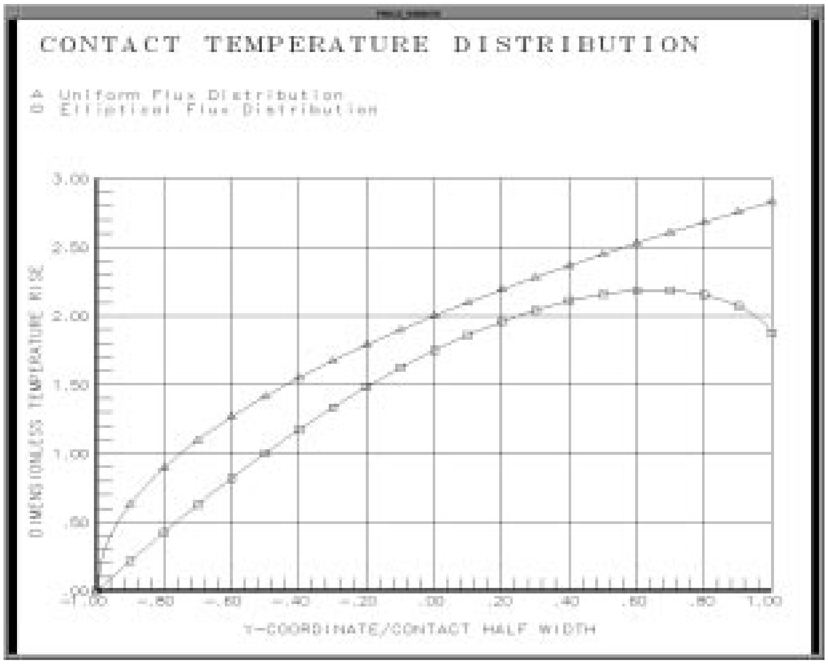

Figure 9. Comparison of temperature distribution resulting from elliptical and unform heat fluxes.

2.10 Overall Boundary Conditions and Model Implementation

Implementation of the above thermal formulation to ADORE will require certain boundary conditions. Basically, the models define a thermal resistance for computation of temperature rise for a given heat generation. Thus it is necessary that temperature at one reference point be prescribed. In the present investigation this temperature is assumed to be the housing temperature. Alternatively, the reference temperature can be shaft or the exit coolant temptress. The inlet temperature of the coolant is of course an input. Thus the steps in thermal model implementation consist of the following:

1. Compute heat generation from bearing interactions.

2. Continue the heat generation computations over a specified time domain and compute an average over the selected time domain.

3. Establish heat partition from prescribed material properties and compute heat transferred to the housing.

4. Compute temperature distribution in the housing and then the outer race.

5. Based on the established outer race contact temperature distribution, compute the temperature at contact with the rolling elements. Temperatures of both the race and rolling element are presently assumed to equal at contact.

6. Now implement the rolling element model to compute the base rolling element temperature and the contact temperature at the inner race.

7. Carry out heat transfer analysis through the inner race and shaft to compute appropriate temperature distribution and bulk temperatures.

8. From overall heat balance the heat transferred to the coolant may not be determined. Now given the flow rate and inlet temperature, the exit temperature may be computed.

9. All of the above is carried out at certain time intervals over which the heat generations are averaged. Clearly, all thermal transients are being neglected. In other words, there is no thermal inertia. However, when the averaging time is large enough such a stepwise implementation is reasonable.

10. Once the temperatures are computed determine the changes in bearing internal geometry if desired and update all geometrical parameters.

11. Apply any constraints, such as equilibrium to maintain constant applied load.

12. Continue numerical integration of equations of motion.

Note that when explicit procedures are used to integrate the equations of motion the above stepwise implementation will not introduce any local truncation. Thus stability or convergence of the integration procedure is unaffected.

3. Other Analytical Models

With the implementation of thermal models provides a dynamic coupling between applied conditions and overall bearing behavior. For the prescribed conditions a realistic model for bearing heat generation and transfer of heat through the bearing, a dynamically changing thermal field may be computed, which becomes a subsequent input to the operating materials parameters and bearing geometry. With such enhancements to bearing performance modeling any imposed constraints, in terms of applied loads and/or displacements, have added practical significance. In particular, equilibrium constraints may be very effectively applied to speed up the dynamic simulations in a rather complex operating environment. For this purpose, enhancements to the equilibrium models in ADORE are developed as a part of the current development. In addition, since the overall modeling approach provides significant improvement in the prediction of actual conditions at the rolling element to race contacts, the models for computing fatigue life are also refined.

3.1 Generalized Equilibrium Formulations

In all prior version of ADORE race equilibrium was restricted to radial and axial force equilibrium. Simulation of applied moments was only possible by prescribed relative misalignment between the races. Although such a formulation is adequate for the simulation of overall load distribution and other quasi-static parameters, it has a limited use for imposing time-varying dynamic constraints involved with moment equilibrium, particularly with angular contact ball bearings with combined thrust and radial loads. An extension of the equilibrium equations to include moment about the two transverse axes is, therefore, the objective of this enhancement.

As shown in figure 10, ball to race interaction is modeled by locating the ball center relative to the raceway.

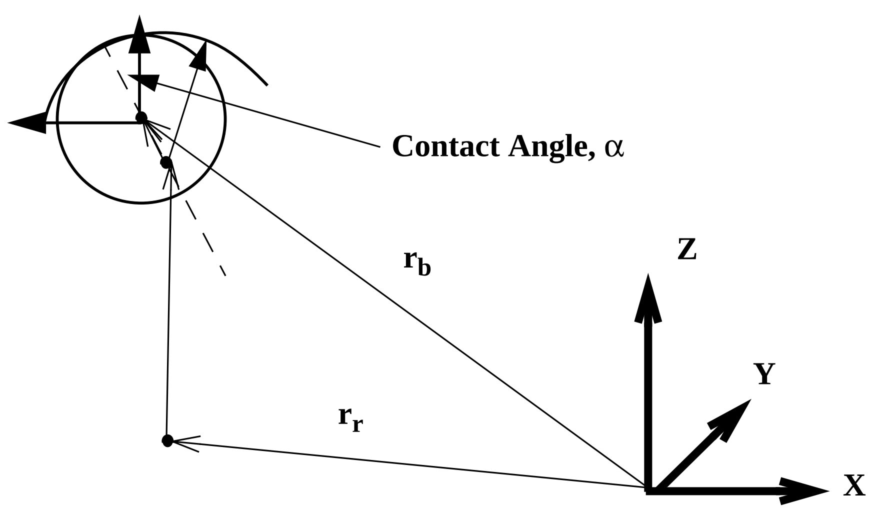

Figure 10. Schematic of ball and race relative position.

Let $\mathbf{r}_b$ and $\mathbf{r}_r$ be the position of ball and race centers in an inertial coordinate frame. When a cylindrical frame is used to track ball position and a cartesian frame is used for prescribing the race position, then these vectors may be written in terms of there components, as

$$\mathbf{r}_b = \begin{bmatrix} x \\ -r\sin\varphi \\ r\cos\varphi \end{bmatrix} \text{ and } \mathbf{r}_r = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{54}$$

and the relative position in inertial frame is simply

$$\mathbf{r}_{br}^{i} = \begin{bmatrix} x \\ -r\sin\varphi \\ r\cos\varphi \end{bmatrix} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{55}$$

For the purpose of computing geometric interaction the above may be transformed to a race coordinate frame

$$\mathbf{r}_{br}^{r} = \begin{bmatrix} T_{ir} \end{bmatrix} \begin{bmatrix} x - X \\ -r\sin(\varphi - Y) \\ r\cos(\varphi - Z) \end{bmatrix} \tag{56}$$

Here $[T_{ir}]$ is the transformation matrix from the inertial to race fixed coordinate frame as defined by the three transformation angles [3]. These angles are rotations, η, θ_2 and θ_3 about the X, Y and Z axes respectively. For the race rotation about X axis simply constitutes race rotation, and when the race is symmetric about the X axis, then this rotation has no effect on any interactions with the rolling elements; the other two rotations represent race misalignment about the two transverse axes. In terms of these three angles the transformation matrix is written as:

$$[T_{ir}] = \tag{57}$$

$$\begin{bmatrix} \cos\theta_2\cos\theta_3 & \cos\eta\sin\theta_3 + \sin\eta\sin\theta_2\cos\theta_3 & \sin\eta\sin\theta_3 - \cos\eta\sin\theta_2\cos\theta_3 \\ -\cos\theta_2\sin\theta_3 & \cos\eta\cos\theta_3 - \sin\eta\sin\theta_3\sin\theta_2 & \sin\eta\cos\theta_3 + \cos\eta\sin\theta_2\sin\theta_3 \\ \sin\theta_2 & -\sin\eta\cos\theta_2 & \cos\eta\cos\theta_2 \end{bmatrix}$$

From the above a race azimuth angle, φ', and a corresponding transformation matrix from the race to race azimuth frame may be defined as follows:

$$T_{ra} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi' & \sin\varphi' \\ 0 & -\sin\varphi' & \cos\varphi' \end{bmatrix} \tag{58}$$

where $\sin\varphi' = \dfrac{-r^r_{br2}}{\sqrt{(r^r_{br2})^2 + (r^r_{br3})^2}}$ and $\cos\varphi' = \dfrac{r^r_{br3}}{\sqrt{(r^r_{br2})^2 + (r^r_{br3})^2}}$

Using the above transformation, the relative position vector may now be transformed to the race azimuth frame and then referenced to a certain reference point for the purpose of computing the geometric interaction. In the case of ball bearings, this reference point is the center of race groove curvature located by axial and radial coordinate values equal to $\bar{X}$ and $\bar{R}$ respectively. The assumption here is that the race is ideally round and these two components are constant for all angular positions. Thus the ball center relative to the race curvature center at any azimuth position is written as:

$$\mathbf{r}^a_{brb} = \left[T_{ia}\right]\left(\begin{bmatrix} x - X \\ -r\sin\varphi - Y \\ r\cos\varphi - Z \end{bmatrix} - \left[T_{ir}'\right]\begin{bmatrix} \bar{X} \\ -\bar{R}\sin\varphi' \\ \bar{R}\cos\varphi' \end{bmatrix}\right) \equiv \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{59}$$

Note that if the load vector and bearing axis (X) are in a plane then in the azimuth frame, the compound a_2 will be zero.

The geometric interaction, δ, is now for a prescribed ball diameter, *d*, and race curvature factor *f*,

$$\delta = \sqrt{a_1^2 + a_2^2 + a_3^2} + d(1 - f) \tag{60}$$

and the contact load is expressed as

$$Q = K\delta^n \tag{61}$$

Using equation (59), and ignoring the out of plane y-component, the contact angle, α, as shown schematically in figure 10, may be computed from the relations

$$\sin\alpha = \frac{a_1}{\sqrt{a_1^2 + a_3^2}} \tag{62}$$

and

$$\cos\alpha = \frac{a_2}{\sqrt{a_1^2 + a_3^2}} \tag{63}$$

The position vector locating the contact point relative to the ball center is simply determined by the unit vector, corresponding to equation (59) multiplied by the ball radius:

$$\mathbf{r}_p = \frac{d}{2}\sqrt{a_1^2 + a_2^2 + a_3^2}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{64}$$

The above formulation is applied to both outer and inner races. The equilibrium equations may then be written for both the balls and races. Using the subscript, *j*, for the races the ball equilibrium equations may be written as summation of the axial and radial component of forces on the outer and inner race contacts. Thus,

$$\sum_{j=1}^{2} Q_j \sin\alpha_j = 0 \tag{65}$$

and

$$\sum_{j=1}^{2} Q_j \cos\alpha_j + F_c = 0 \tag{66}$$

where F_c is the centrifugal force acting on the ball.

The race equilibrium equations may be written in the cartesian inertial frame in terms of a summation of forces acting on all ball/race contacts. Generally, one of the races, normally the outer, is assumed to be fixed while relative position of the other race may be computed by solving the equilibrium equations. In addition to the force equilibrium equations along the X,Y,Z axes, we also have two moment equilibrium equations along the Y and Z axis. Moment equilibrium about the X axis is not applicable, since this is the axis of bearing rotation. Race moment is first computed in the azimuth frame, where the only applicable component is about the y axis. In terms of the various position and force vectors presented above, the moment on the race about the y axis in the azimuth frame may be written as:

$$M_i = \begin{bmatrix} a_{1i} + \bar{X} + \frac{d}{2}\sin\alpha_i \\ 0 \\ a_{3i} + \bar{R} + \frac{d}{2}\cos\alpha_i \end{bmatrix} \times \begin{bmatrix} Q_i \sin\alpha_i \\ 0 \\ Q_i \cos\alpha_i \end{bmatrix} \tag{67}$$

Clearly, the vector moment equation has only one component about the y-axis. The overall force and moment equilibrium equations on the race may now be written as summation over all rolling elements:

$$\sum_{i=1}^{n} Q_i \sin\alpha_i + F_x = 0 \tag{68}$$

$$\sum_{i=1}^{n} Q_i \cos\alpha_i \sin\varphi_i + F_y = 0 \tag{69}$$

$$\sum_{i=1}^{n} Q_i \cos\alpha_i \cos\varphi_i + F_z = 0 \tag{70}$$

$$\sum_{i=1}^{n} M_i \sin\varphi_i + M_y = 0 \tag{71}$$

$$\sum_{i=1}^{n} M_i \cos\varphi_i + M_z = 0 \tag{72}$$

For the above five race equilibrium equations, (68) to (69), the five unknown are the three race center positions coordinates *X*, *Y*, *Z*, and the two relative misalignment angles, θ_2 and θ_3. For the ball equilibrium equations, (65) and (66), the unknowns are the axial and radial position of ball center, *x* and *r*. Clearly, the equations are nonlinear, and an iterative procedure shall be required for a solution. In ADORE a Newton-Raphson procedure is used in two steps. First for a given race position, the ball equilibrium equations are solved. After obtaining the loads satisfying ball equilibrium, iterations for the race equations are performed in the second step. In both steps a Jacobian matrix is required for solution of the simultaneous algebraic equations. The various components of this matrix are obtained by straightforward differentiation of the equilibrium equations.

3.2 Fatigue Life Model

A model for computation of improved fatigue life is based on the recent work done by Tallian [16-17]. Enhancement to life prediction is achieved by applying life mollification factors on the basic Lundburg-Palmgren life, N_{LP}, to obtain the corrected life, N_{10}, with a 10% failure probability. Formulae used in ADORE for the base N_{LP} life of a rolling contact, as used in ADORE, are documented in an earlier publication[21].

$$N_{LP} = \left(\frac{Q_e}{Q_c}\right)^{-p} \tag{73}$$

where Q_e is an equivalent contact load, Q_c is the contact load capacity, and p is a load exponent.

In the case of a ball bearing, the equivalent load is the actual contact load, and the load capacity is defined by an expression [22]:

$$Q_c = \mathrm{A}\left(\frac{2f}{2f-1}\right)^{0.41}(1 \pm \gamma)^{1.39}\left(\frac{D}{d_m}\right)^{0.39} D^{1.80} n^{-1/3} \tag{74}$$

where A is an ANSI fatigue contact [N/m$^{1.80}$], d_m is the pitch diameter [m], D is the ball diameter [m], f is the race groove conformity ration (groove radius/ball diameter), n is the number of stress cycles per race revolution per rolling element, $\gamma = (D\cos\alpha)/d_m$, α being the contact angle.The positive and negative signs in the second term in parenthesis apply respectively to the outer and inner races. Also, the commonly used value for the fatigue constant A is $2.3615 \cdot 10^7$ N/$m^{1.80}$.

For roller bearings, there the contact load intensity may vary significantly along the roller length, the equivalent load is defined in terms of an equivalent load intensity

$$Q_e = 2a\left[\frac{1}{2a}\int_{-a}^{a} q^{pe}dx\right]^{1/(pe)} \tag{75}$$

where a is the contact half width, p is a load exponent and e is the Weibull dispersion slope for the contact under consideration.

Also, the load capacity for a line contact is written as:

$$Q_c = \mathrm{B}(1 \pm \gamma)^{29/27}\left(\frac{D}{d_m}\right)^{2/9} D^{29/27} l^{7/9} n^{-1/4} \tag{76}$$

where B is the ANSI fatigue constant [N/$m^{50/27}$] for line contact, l is the length of contact and all other variables are same as those defined above for ball bearing contacts. The commonly used value for the constant B is $1.98 \cdot 10^8 (N/m^{50/27})$

Life correction factors are now applied on the above computed base life, as formulated by Tallian [16-20].

$$N_{10} = 12AM^{-\frac{1}{\beta}}\left[N_{LP}^{-\frac{1}{\zeta}}\tilde{\Psi}^{\frac{1}{\beta\zeta}} - \tau_o\Phi_{0LP}^{\frac{1}{\beta\zeta}}\right]^{-\zeta} \tag{77}$$

where A is a dimensionless scaling multiplier (=0.23), M is a dimensionless matrix sensitivity multiplier, as tabulated below in table 3, β is a dimensionless life dispersion exponent (=1.60), ζ is a dimensionless stress/life exponent (=7.33), τ_o is the material fatigue limit stress (MPa) with a default (τ_o=0), and Φ_{0LP} is a baseline matrix susceptibility factor ($=3 \cdot 10^{-45}$).

Table 3: Material Matrix Susceptibility Multiplier

Material (Steel Composition)	Matrix susceptibility multiplier M
52100	1.197
8620	1.773
M50	2.267

The equivalent hazard factor ratio $\tilde{\Psi}$ in equation (77) over one duty cycle is defined as:

$$\tilde{\Psi} = \frac{\tilde{\Phi}_T}{\tilde{\Phi}_{TLP}} \tag{78}$$

An expression for equivalent hazard factor product $\tilde{\Phi}_T$ over one duty cycle is written as:

$$\tilde{\Phi}_T = \frac{\beta}{l}\tilde{N}^{-\beta}\sum_{j=1}^{q}\left\{\sum_{k=1}^{m}\left[\Phi_{4,k,j}\sum_{i=a,b,f}\Phi_{1i}\Phi_{2i}\Phi_{3i}\right]\Delta l N_k^{\beta-1}\right\}\Delta N_j \tag{79}$$

Here ΔN_j in a load cycle increment in relative race revolutions,

$$\tilde{N} = \sum_{j=1}^{q}\Delta N_j \tag{80}$$

is the duty cycle in relative race revolutions divided in q increments per duty cycle, Δl is the track increment in radians, m is the number of equal track increments, with $l = m\Delta l$. The number of track increments, m, may be quite arbitrary. This track increment and the total number of track increments shall determine accuracy of the overall computation of the integrated value by using the above summation. In the bearing dynamics code, however, the load values, at any instant

of time, may be available only at the points where the rolling elements are located. Thus it is may be convenient to set *m* equal to the number of rolling elements. This may result in a slight inaccuracy when the number of rolling elements is quite small, but for most practical bearings, such an assumption may be quite acceptable.

The severity balancing factors for asperity and furrow defects are model constants

$$\Phi_{1a} = \Phi_{1f} = 1.50 \tag{81}$$

Also, the severity balancing factor for subsurface defects is a model constant,

$$\Phi_{1b} = 1.0 \tag{82}$$

The asperity defect severity factor Φ_{2a} is given by the relation:

$$\Phi_{2a} = D_s[1 - G(\Lambda - 0.40)]\text{, with } \Lambda > 0.40 \tag{83}$$

while, G() is the standardized Gaussian cumulative distribution function, which may be approximated by the curve fitted relation:

$$\frac{1 - G(x)}{\varphi(x)} = 1.03\{e^{0.75x} + 0.22\} \tag{84}$$

with the Gaussian frequency function,

$$\varphi(x) = \sqrt{2\pi}e^{-\frac{x^2}{2}} \tag{85}$$

The asperity count, D_s in equation (83) is defined as

$$D_s = 0.0306\left(\frac{\sigma}{\theta}\right)^{-2}\alpha \tag{86}$$

with the roughness spectrum width parameter, $\alpha = 40$, σ and θ are respectively the composite roughness height (mm) and slope (radians) for the two interacting surfaces:

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2} \text{ and } \theta = \sqrt{\theta_1^2 + \theta_2^2} \tag{87}$$

In the event no data on roughness height and slope is available, the following empirical relation may be used:

$$\frac{\sigma}{\theta} = 0.02683\sigma^{0.2607} \text{ mm/rad} \tag{88}$$

The defect severity factor, Φ_{2b}, in equation (79) is really the materials processing multiplier:

$$\Phi_{2b} = \Phi_{2bLP} I \tag{89}$$

where $\Phi_{2bLP} = 1$, is a model constant and it represents the Lundberg-Palmgren steel processing factor, and *I* is a materials processing multiplier, depending on the manufacturing process used. The default values are tabulated below in table 4.

Table 4: Steel Processing Multiplier

Processing Method Designation	Method Description	Processing Multiplier, *I*
CVD	Carbon vacuum deoxidation through-hardening steel (groups pre-dating 1975)	2.58
$\text{CVD}_{\cancel{\text{new}}}$	Carbon vacuum deoxidation through-hardening steel (groups 1975 and later)	0.077
CVD_{carb}	Carbon vacuum deoxidation through-hardening steel (all dates)	4.85
VIMVAR	Vacuum induction melt, vacuum are remelt	0.0030

The furrow defect severity factor Φ_{2f} allows for cleanliness of the manufacturing process.

$$\Phi_{2f} = \Phi_{2fclean} C_{cont} \tag{90}$$

Here the baseline factor, $\Phi_{2fclean}$ has a default value of 600, while C_{cont} is a cleanness factor, introduced primarily for future implementation. Presently, there is no significant experimental data to substantiate any value other than the present default of 1.0.

The subsurface stress field factor Φ_{3b} is expressed as:

$$\Phi_{3b} = t_b^{\beta\zeta} \tag{91}$$

with

$$t_b = 0.39 + 0.10 s_{stat} \tag{92}$$

and

$$s_{stat} = \frac{(\sigma_{res} + 0.834\sigma_{hoop})}{p_{max}} \tag{93}$$

where σ_{res}, σ_{hoop} and p_{max} are respectively the residual stress, hoop stress and the maximum Hertzian pressure. All quantities are in stress units.

The stress field factor for furrows, Φ_{3f} is estimated from the relation

$$\Phi_{3f} = t_f^{\beta\zeta} \tag{94}$$

with

$$t_f = 1.25z_{frac} + 1.22\mu_a + 0.15s_{stat} \tag{95}$$

in which $z_{frac} = 0.0270$, and $\mu_a = 0.10$ are the suggested default values for the depth factor and the asperity traction coefficient.

The stress field factor for asperity defects Φ_{3a} is:

$$\Phi_{3a} = t_a^{\beta\zeta} \tag{96}$$

where

$$t_a = 1.25z_{frac} + 1.22\mu_{TOT} + 0.15s_{stat} \tag{97}$$

in which the total traction coefficient, $\mu_{TOT} = \mu_{eff} + \mu_{EHD}$, is the sum of the EHD film traction coefficient, μ_{EHD}, and μ_{eff} is the effective asperity traction coefficient, which is give by:

$$\left(\frac{\mu_{eff}}{\mu_a} = 0.05\frac{E'}{p_{max}}\theta(\alpha - 0.9)^{3/4}F_{32}\right) \le 1 \tag{98}$$

where, $\frac{1}{E'} = \frac{1-\upsilon_1^2}{E_1} + \frac{1-\upsilon_2^2}{E_2}$, E_1, υ_1, E_2, υ_2 being the elastic modulus (Mpa) and Poisson's ratio respectively for the two interacting materials, p_{max} is the maximum Hertz pressure (MPa), θ is the composite asperity slope defined earlier, α was defined earlier in equations (86-87), and the function F_{32} is given by the relation

$$F_{32} = \exp\left[-15.50 + 7.50(4.30 - \Lambda)^{\frac{1}{2}}\right] \tag{99}$$

where Λ is the ratio of lubricant film thickness to the composite asperity height. Note also that when equation (98) yields a value for $\frac{\mu_{eff}}{\mu_a}$ greater than 1.0, then value must be set equal to 1.0. In addition when there is no lubricant film, $\mu_{eff} = \mu_a$, and the EHD traction coefficient, μ_{EHD} is set to zero.

The load distribution factor, Φ_4, in equation (79), is identical to its Lundberg-Palmgren value and it is given by:

$$\Phi_4 = \Phi_{4LP} = -0.39^{-\beta\zeta}\ln(0.90)\Phi_{0LP}^{-1}N_{LP}^{-\beta} \tag{100}$$

When the factors are substituted in equation (78), the constant part of the above equation will cancel out; thus for numerical implementation the only relevant part is $N_{LP}^{-\beta}$.

In addition, the Lundberg-Palmgren hazard factor product, $\tilde{\Phi}_{TLP}$, in equation (78), is a special case of equation (79), with the asperity (subscript *a*) and furrow (subscript *f*) defect products set to zero, with equations (82), (89), $\Phi_{1bLP} = \Phi_{2bLP} = 1$, and with equations (91) and (92), $\Phi_{3bLP} = 0.39^{\beta\zeta} = 1.60 \cdot 10^{-5}$,

$$\tilde{\Phi}_{TLP} = 0.39\frac{\beta}{l}\tilde{N}^{-\beta}\sum_{j=1}^{q}\left\{\sum_{k=1}^{m}\Phi_{4LPkj}\Delta l N_j^{\beta-1}\right\}\Delta N_k \tag{101}$$

Now the equations for both $\tilde{\Phi}_T$ and $\tilde{\Phi}_{TLP}$ are computed for both races independently and they are applied in equations (78) and (79) to compute modified life of each race independently. These computed lives are then combined to obtain the composite bearing life using the expression:

$$N_{10}^{-e} = N_{10OR}^{-e} + N_{10IR}^{-e} \tag{102}$$

Since the formulae for computing the life modifying factors involves both variation with load and time, all computations are first performed for one relative race rotation. Normally, in a practical bearing one of the races rotates with respect to the applied load, while the other is stationary. For the rotating race each contact will be subject to the same load condition while for the stationary race each contact will be subject to a different load condition, however, there will be no variation with time in this case. Now for the rotating race, and for a load cycle comprising of one

revolution, we can make the following substitutions in equation (79):

$$l = 2\pi,\ N_k = \frac{2\pi}{m}k,\ \tilde{N} = 2\pi \text{ and } \sum_{j=1}^{q} \Delta N_j = \tilde{N} = 2\pi \tag{103}$$

Thus the expression for $\tilde{\Phi}_T$, as stated in equation (78) becomes:

$$\tilde{\Phi}_{TR} = \frac{\beta}{m}(2\pi)^{1-\beta} \sum_{k=1}^{m} \Phi_{TRk}\left(\frac{2\pi}{m}k\right)^{\beta-1} \tag{104}$$

Note that since all contacts are subject to the same load condition the terms under the outer summation are constant. Therefore, the summation $\sum_{j=1}^{q} \Delta N_j$ may be written as a constant with a value equal to 2π. Also, the subscript R is used for the rotating race, and the product of various life modification factor is denoted by Φ_T as:

$$\Phi_{Tk} = \Phi_{4k} \sum_{i=a,b,f} \Phi_{1i}\Phi_{2i}\Phi_{3i} \tag{105}$$

For the race which is stationary with respect to the load, each contact is subject to different load conditions but there is no variation in time. Thus in equation (79) the time and space integrals may be separated to yield:

$$\tilde{\Phi}_{TS} = \frac{\beta}{(\beta-1)m} \sum_{k=1}^{m} \Phi_{TSk} \tag{106}$$

where the subscript S is used to denote stationary race.

Corresponding to equations (104) and (106) the corresponding Lundberg-Palmgren factors are written as:

$$\tilde{\Phi}_{TLPR} = \frac{\beta}{m}(2\pi)^{1-\beta} \sum_{k=1}^{m} \Phi_{TLPRk}\left(\frac{2\pi}{m}k\right)^{\beta-1} \tag{107}$$

$$\tilde{\Phi}_{TLPS} = \frac{\beta}{(\beta-1)m} \sum_{k=1}^{m} \Phi_{TLPSk} \tag{108}$$

and

$$\Phi_{TLPk} = \Phi_{4LPk} \sum_{i = a, b, f} \Phi_{1LPi}\Phi_{2LPi}\Phi_{3LPi} = \Phi_{4LPk}\Phi_{2bLP}\Phi_{3bLP} \tag{109}$$

In a bearing if both races rotate relative to the applied load, then the expressions for rotating race are used for both races.

Implementation of the life model is now straightforward. After performing the load computation at each outer and inner race contact, carry out the following steps to compute the corrected life:

1. Use equations (73) to (76) to compute the base Lundberg-Palmgren life for each contact.

2. Compute the life modification products, Φ_{Tk} and Φ_{TLPk} at each contact using equations (105) and (108) respectively.This will of course implement all the other equations, (79) to (101) for computation of the various factors.

3. Use equations (104) and (107), if the race is rotating, or (106) and (108), when the race is stationary, to compute the integrated factors $\tilde{\Phi}_T$ and $\tilde{\Phi}_{TLP}$ for each of the races.

4. Use equation (78) to compute the overall factor life modification factor $\tilde{\Psi}$ for each of the races.

5. Perform appropriate summation to computer the base Lundberg-Palmgren life for each raceway, N_{LP}.

6. Implement equation (77) for each of the races to compute the modified life. Note the fact that τ_o is in units (MPa), while the numerical value of Φ_{0LP} is $3 \cdot 10^{-45}$. Depending on the precision of floating point numbers, direct implementation of these values may result in some numerical problems. It may, therefore, be more convenient to input stress in units (Pa) and prescribe the value of $\Phi_{0LP}^{1/(\beta\zeta)}$. This will reduce the second terms in equation (77) to $\tau_0 \cdot 1.5985 \cdot 10^{-10}$, which may be numerically more manageable.

7. Apply equation (102) to combine the base and modified lives of the races to obtain the base and modified life of the entire bearing.

4. Enhancement to Bearing Dynamics Code ADORE

All analytical models during this project are implemented in the bearing dynamics computer code, ADORE. In addition to the model enhancements, the code is basically rewritten to conform to the FORTRAN-90 standard. It is expected that this new code architecture will be better understood by the users and it will be more amiable to future enhancements and refinements.

4.1 Conversion to FORTRAN-90 Standard

ADORE when first published in 1984 [3] was written in FORTRAN-77, which at the time was the most common language used for the development of engineering applications. As the modeling technology advanced, ADORE was continually updated to incorporate new models and improvements in overall modeling technology. Both due to the complex nature of the models and limitations of FORTRAN-77, the code architecture became continually complicated until it reached a point where it became extremely difficult for a user to comprehend. This prompted conversion of the code to FORTRAN-90 standard which gained popularity in the 1990's.

With regard to ADORE and other complex engineering applications, FORTRAN-90 offers several programming enhancements:

1. Completely "top-down" design free of any statement labels and "go-to" statements.
2. Descriptive variable names.
3. Array operations for efficient array processing and vector operations.
4. Data modules for efficient data handling.
5. Internal procedures for better code encapsulation.
6. User defined precision for consistent computational accuracy.

The above are only some of the features used by ADORE. On the whole FORTRAN-90 comes very close to other more popular languages, such as C, C++ and Java, in terms of code architecture, although FORTRAN-90 is not a object oriented language. Perhaps, the next step in code enhancement will be in terms of implementation of Java tools and techniques. It is anticipated that the FORTRAN-90 conversion, undertaken in the current project, will greatly facilitate such future developments.

Although there are a number of application conversion software packages available to convert old FORTRAN codes to the new FORTRAN-90 standard, the output from these convertors do not provide any simplifications in code architecture and variable descriptions. The current effort was, therefore, directed toward essentially rewriting the code to make use of the new features available in the FORTRAN-90 standard. As shown schematically in figure 11, the newly converted FORTRAN-90 consists of data modules, and both internal and external procedures. In addition the input and output facilities provide efficient user interfaces for preparation of input data and review of program output.

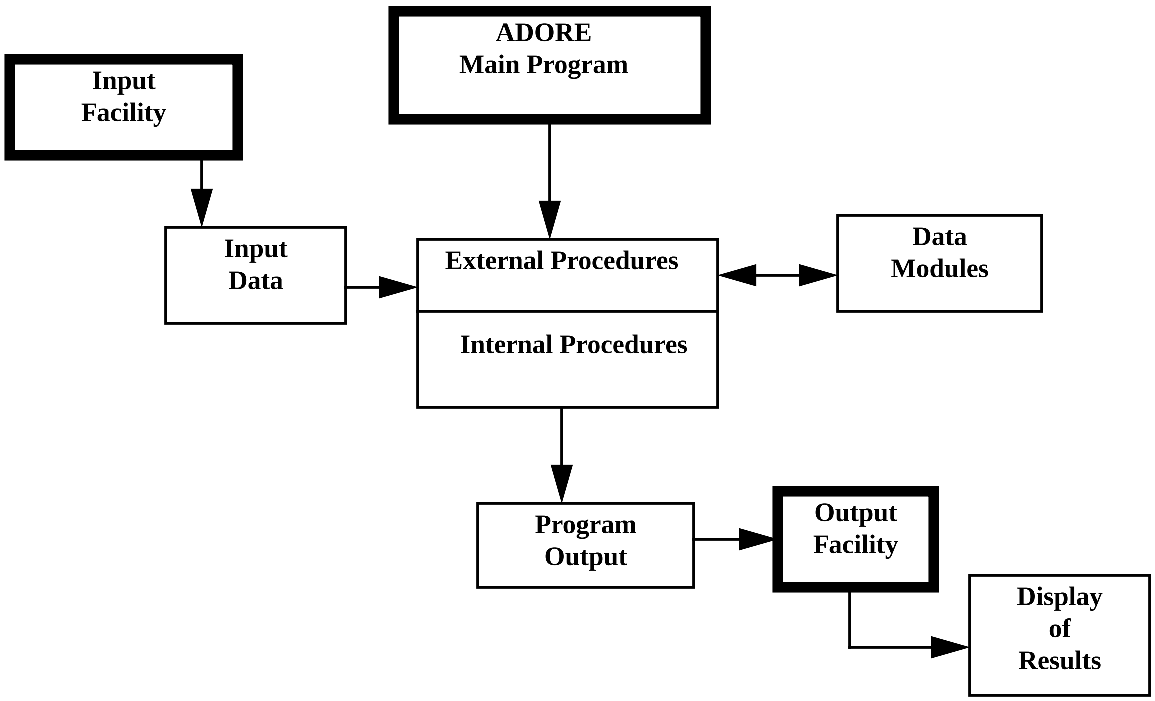

Figure 11. Schematic outline of code elements of ADORE

The development process consisted of first encapsulating all data in data modules with new descriptive variable names and related documentation. A typical data module is listed in Appendix A. This module encapsulates all data pertaining to bearing geometry. Next, the code in each of the compute modules was rewritten in the form of external procedures or subroutines, with new variable names and significantly improved code architecture. All statement labels and “go-to” statements were eliminated to obtain a truly “top-down” design. This permits easy code readibility. The numerical processing procedures uniquely required by the external procedures were incorporated as internal procedures within the given external procedure. A typical code segment is documented in Appendix B. This procedure computes the rolling element to race interaction and contact load. First the rolling element is located relative to the race, then depending on the geometry of elements appropriate geometrical interaction is computed. While a single computation is adequate for a point contact problem, the line contact computations are divided in two parts, first computation of overall contact length and then computation of geometrical interaction for the various grid points within the contact zone. Then appropriate constitutive relation is called for computation of contact load and other contact parameters. In the case of a roller bearing, the roller end to guide flange contact is also contained in this subprogram. Clearly, the code can be easily read from top to bottom without any complicated nesting.

4.2 Procedures for Thermal Interactions

All analytical modules and numerical procedures outlined in section 3 of this report are implemented in ADORE. New subprograms for both external and internal procedures, along with appropriate data modules are developed in a new program group for thermal interactions. Consistent with the naming convention used in ADORE, a new program group ADRH is introduced. The main procedure ADRH1 takes the relevant material properties, prescribed boundary conditions, and computed heat generations as input, to perform the overall heat balance and compute the temperature field through the bearing. Computation of temperature rise, module ADRH2, in the contact is directly related to rolling element to race traction and heat generation. It is therefore, called directly by the rolling element to race traction models. Presently, the contact temperature rise does not feed back to the traction model due to model limitation in traction model limitations for thermal interactions. Such a complex interaction is presently deferred to future development. The program module ADRH3 carries out all the convection analysis to provide appropriate heat transfer coefficients, and this is interactively called by the main module ADRH1 as a function of computed temperatures. Thus the heat transfer coefficients may vary as a function of temperature. The overall compute module structure is illustrated in the simplified diagram shown below in figure 12.

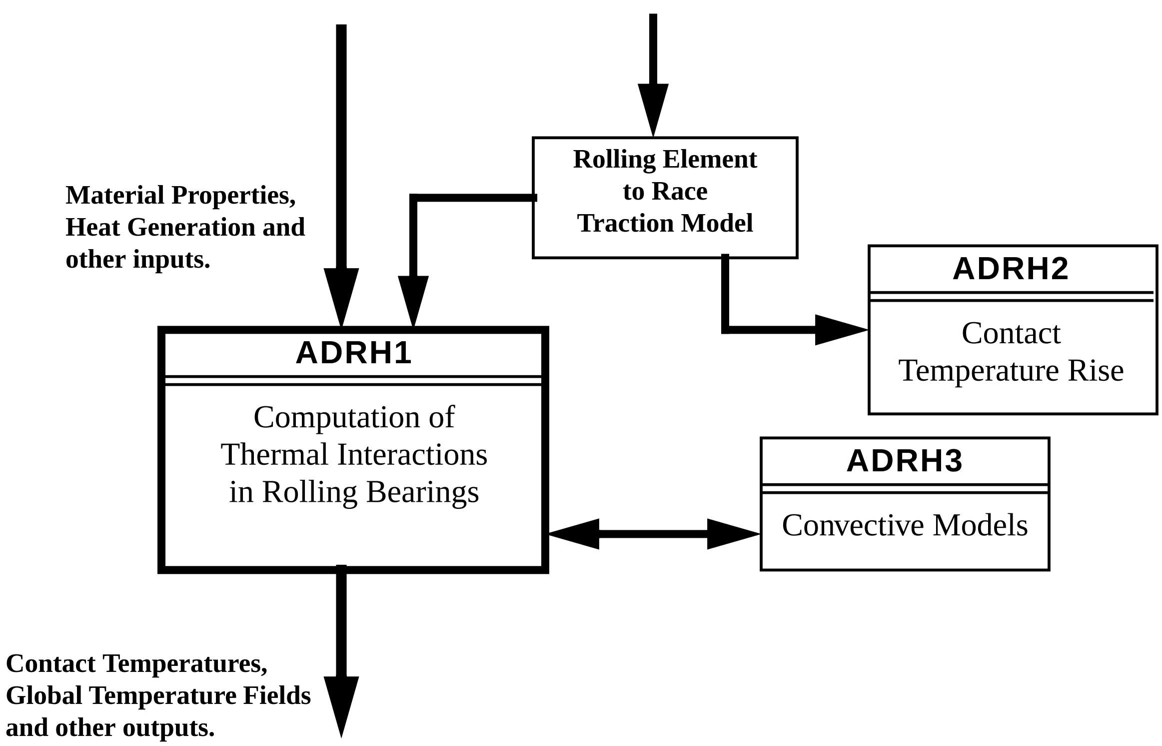

Figure 12.Schematic architecture of the thermal module in ADORE.

The above thermal interaction modules are first debugged independently to ensure proper operation and then incorporated in ADORE as a new program group. In its current form, ADORE consists of modules A to G, as shown schematically in figure 13. All module names start with ADR, abbreviation for ADORE, following by a group letter A, B, C, etc., and then followed by a

numerical digit indicating the sequence number of the procedure. Module ADRH, shown in the shaded box in figure 13, is the outcome of the current effort. This module is primarily called by the traction module ADRD, where all details of contact geometry and load conditions are available. The modules may also be called by the main program, any time when an estimate of current temperature fields is desired. Module inputs, e.g., material properties and computed heat generation, are provided by appropriate data interfaces. Similarly, the computed contact temperatures and other thermal parameters are output via pertinent data structures.

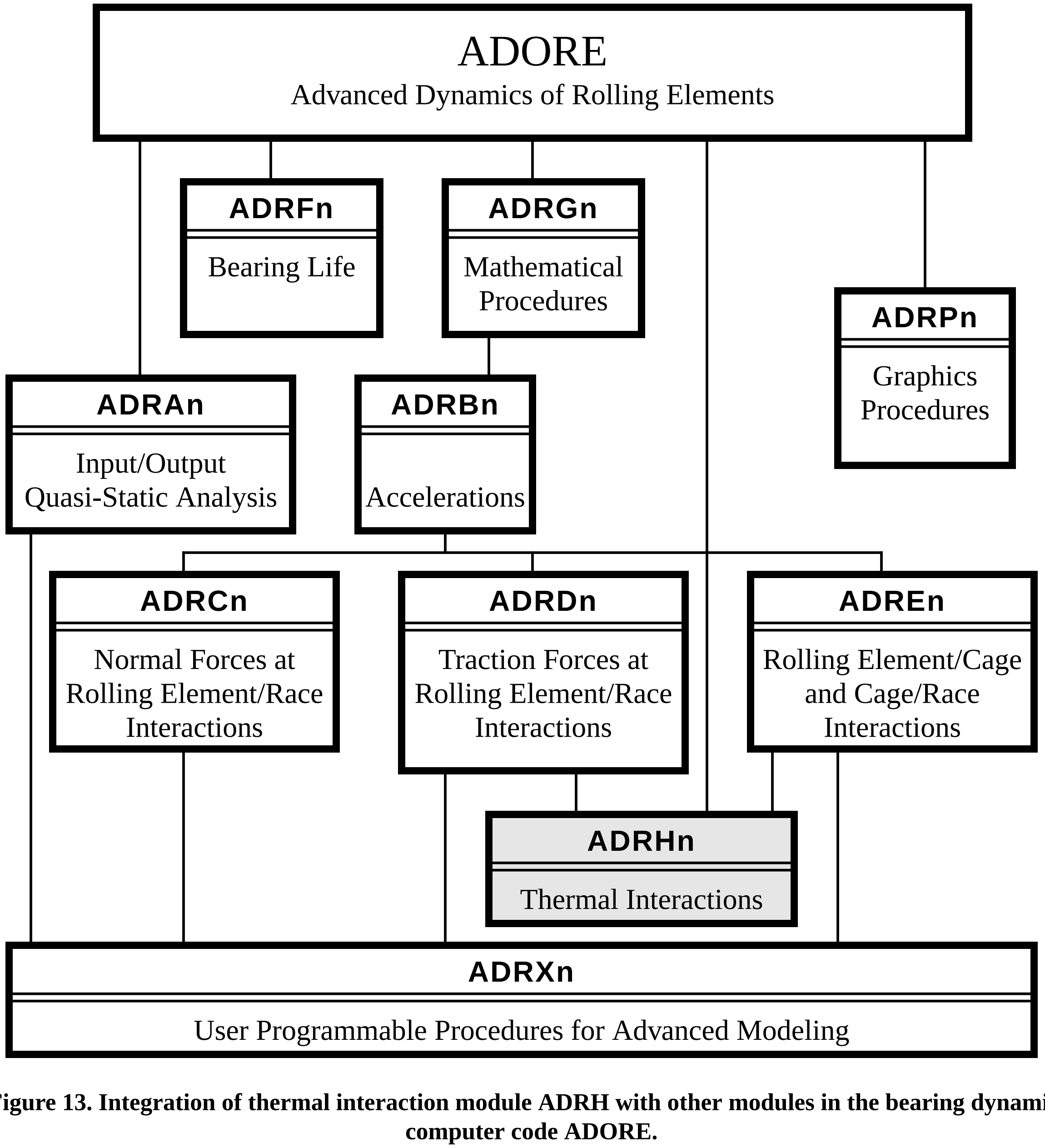

Figure 13. Integration of thermal interaction module ADRH with other modules in the bearing dynamics computer code ADORE.

4.3 Implementation of Time-varying Geometry

Although not shown, for simplicity, in figure 13, the computed thermal field interacts with the rolling element to race interaction module responsible for the computation rolling element to race geometrical interaction and normal load. The interaction is provided via the thermal distortions module, ADRC2. Here the computed temperatures by ADRH1 are taken as input to compute the change in element geometry and internal clearance in the bearing, which in turn is fed as input to the normal load computation. Thus the thermal interactions affect the normal load computation, which in turn will alter traction and, therefore, the heat generation, which is fed back to temperature computation. Thus the computation appears to be an iterative loop. However, in view of transient effects associated with the thermal problem, which are presently not considered, such an iterative interaction may not be fully justified. Thus implementation in ADORE is such that the temperatures computed at a given step are used to compute thermal interaction at the next step. Hopefully, if the step size is small enough and the integration is convergent, the final temperatures will stabilize to some steady state.

It should be noted that the above interaction between the thermal and mechanical effects introduces a fairly sophisticated level of complexity in modeling of bearing dynamics. Before extensive use of such a feature, adequate experimental validation of the models used in ADORE is necessary. In absence of such a validation, the results must be viewed with some caution. The input options stream in ADORE, is therefore, structured to permit computation of the temperature fields without feeding it back to change bearing element geometry. This option may be parametrically used to determine the significance of thermal interactions in overall bearing performance.

4.4 Other Refinements

Other refinements to ADORE, carried out as a part of the current effort, include generalization of equilibrium module for ball bearings and enhancements to the life module as outlined in section 3 of this report. In addition, there have been a number of other enhancements to ADORE, which were carried out as a part of an ongoing commercial effort in parallel with this effort. All of these enhancements are also part of the current version of ADORE. Some of the major improvements are listed below after describing the moment equilibrium and life modeling work carried out as a part of this effort:

4.4.1 Moment Equilibrium for Ball Bearings

Two new moment equilibrium equations for the race have been introduced to perform moment equilibrium about the two transverse axes. Thus a total of five degrees of freedom are now available for equilibrium solutions. This resulted in a complete rewrite of the race equilibrium module which consists of the equilibrium equations and associated Jacobian matrix for solving the equations by Newton-Raphson techniques. All of these enhancements are in the subprogram ADRA4. Since the number of algebraic equations now becomes five, a new subroutine ADRG0 for the solution of a set of algebraic equations is introduced. Earlier the solutions were obtained within the quasi-static subroutine ADRA4. Also, in order to track the very small rotations, resulting from race misalignment, the transformation matrices with small angles are now computed by using series expansions of the sine and cosine functions. In addition appropriate procedures to compute the derivative of these matrices for accurate computation of Jacobian matrices are also incorpo-

rated. Appropriate changes to input data module are also carried out to execute the newly introduced degrees of freedom in the equilibrium formulation. These enhancements is presently available for ball bearings only. The code has not yet been fully developed for roller bearings. Since roller bearings involve flange interactions when the roller are misaligned and/or skewed, convergence of moment equilibrium equations becomes more complicated; particularly due to discontinuities resulting from the rollers getting in and out of flange contact during the iterative process. Thus roller bearings should be run with prescribed misalignment rather than moment.

4.4.2 Fatigue Life Model Refinements

The fatigue life correction factors outlined in section 3.2 of this report are incorporated in the subprogram ADRF1. The old life mollification factors are replaced by the newly developed formulae documented in this report. Implementation of the various algebraic equations is quite straightforward. Appropriate options in the input data are also introduced to invoke the newly introduced materials factors.

Although computation of fatigue life does not interact in any way with the dynamics modeling, the output is useful for practical design. Since all the contact loads and related parameters are collected from the main computing stream of ADORE, the effects of thermal interactions on contact loads are automatically taken care of. Thus in addition to the newly developed life improvement factors, the current version of ADORE includes the effect of thermal interactions on life when appropriate thermal options are invoked.

4.4.3 Partition of Rolling Element to Inertial Frame Transformation

With particular emphasis on accurate modeling of roller misalignment and skew in tapered roller bearings, the roller to inertial frame transformation is broken up into two parts: a base transformation for nominal position, and second, a time dependent transformation as a function of roller misalignment and skew. Thus roller misalignment angle excludes the nominal tilt of the roller axis as applied to tapered roller bearings. Such a partition of the transformation matrices provides improvements in precision in modeling tapered bearings. The enhancement is insignificant for other bearings.

4.4.4 Cage Initial Conditions

For a large number of simulations obtained over the last many years, it has been found that the most stable motion of the cage is when the mass center whirl velocity is equal to its angular velocity. In addition to the published literature, numerous steady-state solutions obtained by ADORE have confirmed such a behavior. Based on these findings, such a whirl behavior is now prescribed as an initial condition. Thus if there are no cage interactions, the cage will whirl in a well defined circular orbit, with the prescribed radius. The orbit may deviate a bit, as a result of cage contacts, but it will basically retain the general shape in case of a stable motion. Under unstable situations, however, the orbit will disintegrate into an erratic motion as a function of time. Thus this enhancement provides a greatly improved diagnosis of cage stability. In earlier versions, only a tangential velocity corresponding to the above whirl was applied at zero time, the remainder of the whirl motion was left up to the cage interactions and overall dynamics. Although such an initial condition will also lead to stable, almost circular, whirl orbits, the time required to reach

such a condition may be significantly longer. Note that under stable conditions the steady-state solution does not depend on the initial conditions. It is only the simulation time, required to reach steady-state, which is affected by the initial conditions.

4.4.5 Simulations of Large Number of Time Steps

For low speed bearings, the steady state solutions normally require simulations over a large number of time steps, where stability of integration becomes strongly dependent on control of numerical precision. With the use of “kind” parameter offered in FORTRAN-90, precision of numerical computation is tightly controlled in ADORE. As a result stable integration of the differential equations of motion is possible overall several hundreds of thousands of time steps. A number of runs recently made on slow speed ball bearings have confirmed such a conclusion. It is also anticipated that with such a control of numerical precision within the program, simulations over millions of steps may now be possible. Of course, such investigations will require the fastest possible processor speeds.

When making runs over a large number of time steps, control of output data is another issue. To effectively address this problem, the notion of substeps, within a step, is introduced. This features simply introduces several substeps within a nominal step, while maintaining the required local step size for integration with a prescribed precision. However, neither the output is processed nor saved over the sub steps. Thus a run over 20,000 time steps, with 10 substeps within a step will result in integration over 200,000 time steps. This feature, along with earlier options of data control, provides fairly efficient management of output over large a number of time steps.

4.4.6 Arbitrary Constraints of Degrees-of-freedom

A new option to constrain the degrees-of-freedom on both rolling elements and cage has been implemented. This permits fairly fast integration of bearing under simplified conditions. For example, in the case of a perfectly aligned cylindrical roller bearing with purely radial load, the degrees of freedom corresponding to axial mass center motion and transverse rotation of both the roller and cage may be suppressed, to produce a relatively fast integration. With the implementation of this option, the need for MODE=3, as used in an earlier version, is eliminated. Thus, in the current version of ADORE, the variable MODE may only have values in the range of -2 to 2.

4.4.7 Initial Ball Angular Velocities

In all earlier versions of ADORE initial ball angular velocity in angular contact ball bearings was set by using the “race control” hypothesis where relative ball spin is permitted on only one of the races. Again, by a number of simulations obtained over the years, it is found that this hypothesis does not really hold for well lubricated bearings, instead the ball angular velocity vector lines up to be essentially parallel with the shaft axis. Some recent investigations in this area have shown that the orientation of the ball angular velocity vector is better determined by minimizing the energy dissipated in the ball/race contacts. Thus in the current version of ADORE, a new option for computing the initial orientation of the ball angular vector by minimizing the energy dissipated in the contacts is introduced. The procedure consists of computing the energy dissipation in the outer and inner race contacts as a function of the orientation of ball angular velocity vector. The point of minimum energy is then determined by interpolation. Due to these

additional computations, the startup computing time at step zero is somewhat larger than that required with the race control options used earlier. However, such an initial condition, once again provides faster convergence to steady-state in comparison to the use of race control hypothesis. An input option to use the race control theory, whenever necessary, is, however retained.

4.4.8 Geometrical Imperfections in Modeling Cage/Race Interactions

In all the earlier versions of ADORE, although extensive geometrical imperfections were permitted in the cage pockets, treatments at the guide lands has always been based on ideal geometry. In practice, when the cage, or the interacting race surface, is out-of-round, there are two new problems: (1) the cage can contact the race at more than one point and cause a “lock-up” situation, and (2) contact with the race, on which is cage is not guided, may also become possible. In order to address these situations, the current version implements the following enhancements:

(a) Cage/race interaction module is modified to compute the derivative of geometrical interaction with respect to the angular position, and then determine the number of points where the derivative is zero, or it changes sign. These points define the potential contact points. Therefore, more than one contact point may be easily modeled.

(b) The number of guide lands is now an arbitrary input; the value can be greater than 2. This permits modeling of geometric interaction at the non-guiding race.

(c) ADORE input includes three options for geometric imperfections at the guide lands: (1) elliptical geometry of cage, race or both, (2) sinusoidal variation of radius of cage or race surfaces, and (3) arbitrary geometrical profiles of both the cage and race guide surfaces.

Although all output generated under the above contact conditions is documented in the print output, the plot output is still restricted to the two guide lands which have dominant interactions. In case of multiple contacts, the contact with a larger force is selected for plotting. These restrictions will be lifted in the next revision of the plot program. In the meantime, the option for arbitrary monitoring of output data, in user subroutine ADRX9, may be used and the data can be plotted by using any data processing programs, such as Microsoft Excel.

4.4.9. Arbitrary Traction Data

Very often the user may have experimental traction/slip data obtained at the bearing operating conditions. In such a case it is now possible to compute the traction coefficient by interpolating this data, thus eliminating any uncertainties associated by first computing regression coefficients for a given family of traction curves. This is done in the new user subroutine ADRX7.

The cage geometrical imperfections, specified earlier in ADRX7, are now modeled in ADRX8 and the optional output monitoring is moved to ADRX9.

4.4.10 Moving Reference Frames

A new flag kRotFrames is introduced on input record 3.4 to simulate the effect of moving reference frames as encountered in bearing applications in planetary systems and crank shafts of reciprocating engines. This flag permits rotation of the base bearing frame with a constant veloc-

ity and a fixed radius; these inputs are supplied on new data record 9.4. Again the new inputs have been added to the input facility ADRIN.

4.4.11 Cylindrical Pockets for Roller Bearings

A new cage pocket code, kPocType=-1, has been introduced on record 7.0 to implement cylindrical pockets for roller bearings. This permits simulation of roller guided cages with cylindrical pockets. The output contact angles are included in the plot data files to precisely define position of contacts. Minor modifications to the plot program ADRPB have been made to plot these angles.

4.4.12 Step Size Control During Cage Contacts

When the size of time step is large, either under prescribed inputs or as a result of dynamic step size control, the cage pocket interactions may become quite large or sometimes they may even go out of bounds. To control such an occurrence and model the cage interaction more accurately two enhancements have been made in the current version:

1. The rolling element to cage contact stiffness is refined to make it equivalent to conventional Hertzian contact. This has resulted in a reduction in step size corresponding to an increase in contact stiffness. However, the simulation of rolling element to cage collision if significantly enhanced.
2. In the event of an abnormal contact, an internal error flag is introduced in the rolling element to cage interaction procedure. This flag is seen by the numerical integration procedure, which rejects the solution and starts the integration again with a reduced step.

A combination of the above refinements has added a significant precision to the simulation of rolling element to cage contacts. Such an enhancement is particularly significant when there are a large number of cage pocket interactions, either due to reduced clearances or in the event of an instability. However, these improvements have resulted in a reduction in permissible step size.

4.4.13 Enhancement of ADRX1 Interface

Output as a function of time may now be controlled in the user programmable subroutine ADRX1. The output may be written in any user created data file and later interfaced with other applications and graphics packages. This is particularly helpful when the transient response from ADORE has to be either compared with or implemented in other dynamic models.

In default mode ADORE makes use of symmetry of the bearing and applied loads to save computing effort. However, a generalized programming in ADRX1 may sometimes violate the symmetry rules and thereby produce errors in the solutions. To eliminate such a problem, a symmetry switch kSymetry may now be turned on and off within ADRX1. This switch is stored in the data module SubX, which is generally used in the user programmable subroutines.

4.4.14 Applied Loads and Moments Output

The applied loads and moments in the print output are now divided into two parts: the externally applied loads and moments, and the forces and moments generated internally by the rolling

element to race contacts. This helps in better understanding of race accelerations in the dynamic mode. In equilibrium mode, all forces and moments, of course, sum up to zero.

5. Results

After incorporating the program enhancements, outlined in the preceding sections, several test runs were made to demonstrate the enhanced capabilities. Both ball and cylindrical roller bearings are used to perform these test runs. Bearing geometry and the imposed operating conditions are discussed below before presenting the respective results.

5.1 Test Ball Bearing

A typical gas turbine engine 100 mm bore angular contact ball bearing is used to carry out the ball bearing runs. Bearing geometry and operating conditions are presented below in table 5. Actual input data for ADORE and typical print output for the current bearing are documented in Appendix C.

Table 5: Ball Bearing Geometry and Operating Conditions

Bearing Bore (m)	0.100	Lubricant Type	MIL-L-7808
Bearing Outside Diameter (m)	0.180	Traction Model	EHD
Ball Diameter (m)	0.01905	Cage Friction Coefficient	0.050
Number of Balls	18	Applied Thrust Load (N)	4,500
Pitch Diameter (m)	0.140	Applied Radial Load (N)	2,000
Free Contact Angle (deg)	25.00	Shaft Speed (RPM)	20,000
Outer Race Curvature Factor	0.52	Room Temperature (K)	323
Inner Race Curvature Factor	0.54	Outer Race Fit (m)	0.000010
Cage Outer Diameter (m)	0.146292	Inner Race Fit (m)	0.000050
Cage Inner Diameter (m)	0.130073	Coolant Type	Lubricant
Cage/Race Guidance Type	Inner	Coolant Inlet Temperature (K)	323
Cage Width (m)	0.036241	Coolant Flow Rate (m**3/s)	0.000010
Cage Guide Land Clearance (m)	0.0015		
Cage Guide Land Width (m)	0.0080		
Cage Pocket Clearance (m)	0.0012		

5.2 Moment Equilibrium for Angular Contact Ball Bearings

When an angular contact ball bearing is subjected to a combined thrust and radial load, then the contact loads on the balls vary around the bearing. As a result a moment is exerted about the

transverse axis. If one of the races is free to move, then the race will misalign to satisfy moment equilibrium. The added moment equilibrium equations in ADORE, permit computation of this misalignment and appropriate change in load distribution. The load distribution solutions obtained by performing such a moment equilibrium are shown in figure 14 and 15 for the outer and inner races respectively. Such a distribution is a result of race misalignment about the transverse axis and it will affect the overall ball motion in the bearing.

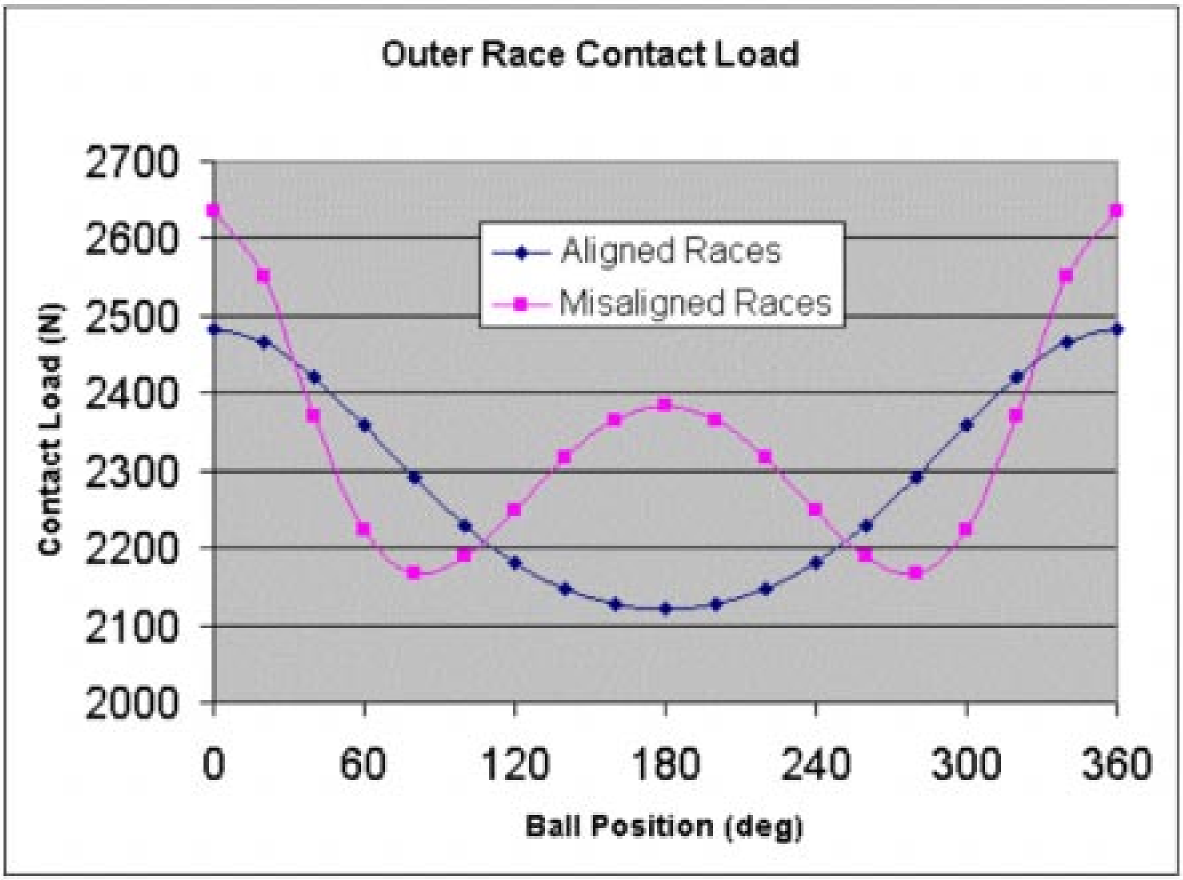

Figure 14. Ball/Race load distribution at the outer race contact with aligned and misligned condition resulting from moment equilibrium.

The figures compare the solutions obtained under a perfectly aligned condition to those obtained under a misalignment computed from the moment equilibrium condition.

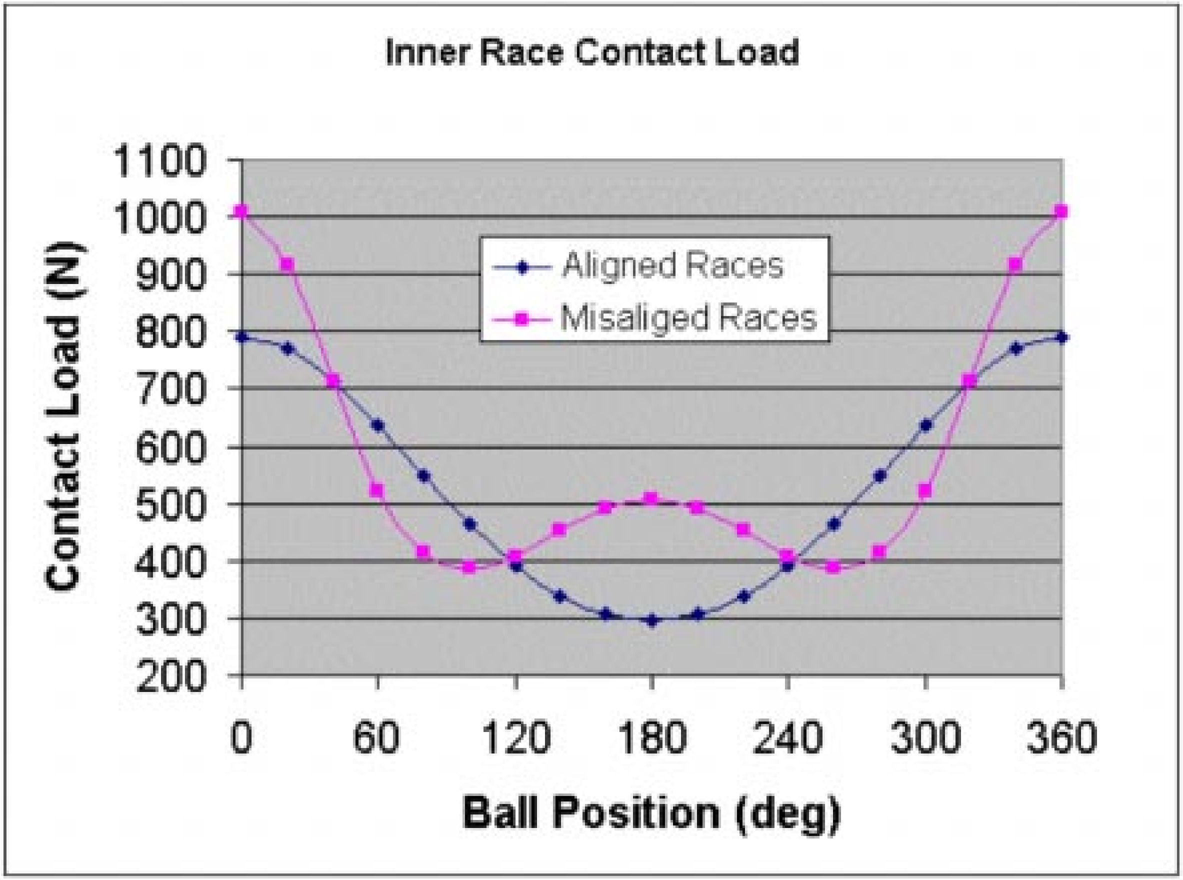

Figure 15. Ball/Race load distribution at the inner race contact with aligned and misligned condition resulting from moment equilibrium.

5.3 Thermal Interactions in Ball Bearings

Thermal analysis is carried out in two parts. First the bearing simulations are obtained without any modifications to the bearing geometry resulting from the change in temperature of the various bearing elements. Since the thermal model in ADORE does not perform any thermal transients, the heat generation at the various interactions is averaged over a prescribed time and then a new temperature field is computed. Thus the temperatures are updated at selected times. In the present example the heat generations are averaged over every ten shaft revolutions. The overall results of the simulation obtained under the conditions outlined in table 5 are shown in figures 16 to 23.

Figure 16 shows the variation in total heat generation or powerloss in the bearing. After the initial transients the heat generation stabilizes to a fairly constant value. The overall mechanical

interaction between the bearing elements is indicated by the time-averaged wear rate plotted in figure 17. As the simulation reaches steady-state these rates stabilize to a well defined constant value. Overall bulk temperatures as computed by the thermal models are shown in figure 18, while details of a typical ball/race interaction are shown in figure 19, which shows the local contact heat generation, the temperature rise in the contact, and the estimated race contact temperatures. The race contact temperatures are output of the thermal model where the local heat generations are averaged over a shaft revolution. Note that the race contact temperatures, along with other bulk temperatures shown in figure 18, change in steps. This is a result of the thermal averaging procedure.

Cage motion is shown in figure 20 to 23. Figure 20 shows the variation in cage mass center velocity. Clearly, the cage whirls at a fairly constant velocity with practically no radial velocity. The resulting cage/race forces are shown in figure 21, which shows that after the initial transient there is little or no contact at the guide land. This is also confirmed by the orbit radius being less than the cage/race clearance as shown in figure 22. The dotted circle in figure 22 denotes the clearance circle. Typical interaction in cage pockets is shown in figure 23. The balls occasionally contact the cage pockets over very short times. This is typical of cage pocket contacts in most bearings.

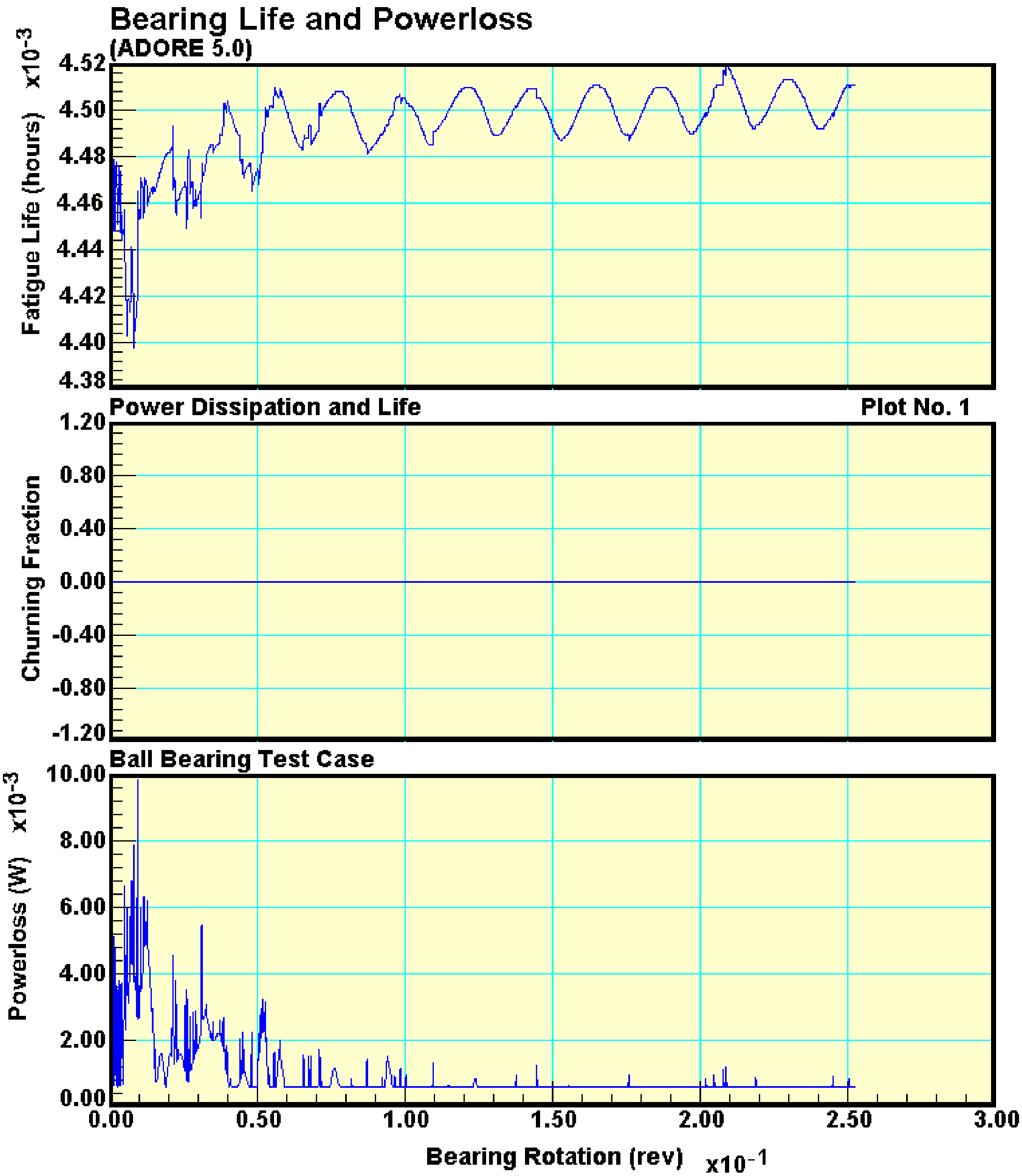

Figure 16. Overall powerloss in the test ball bearing with no thermal interactions.

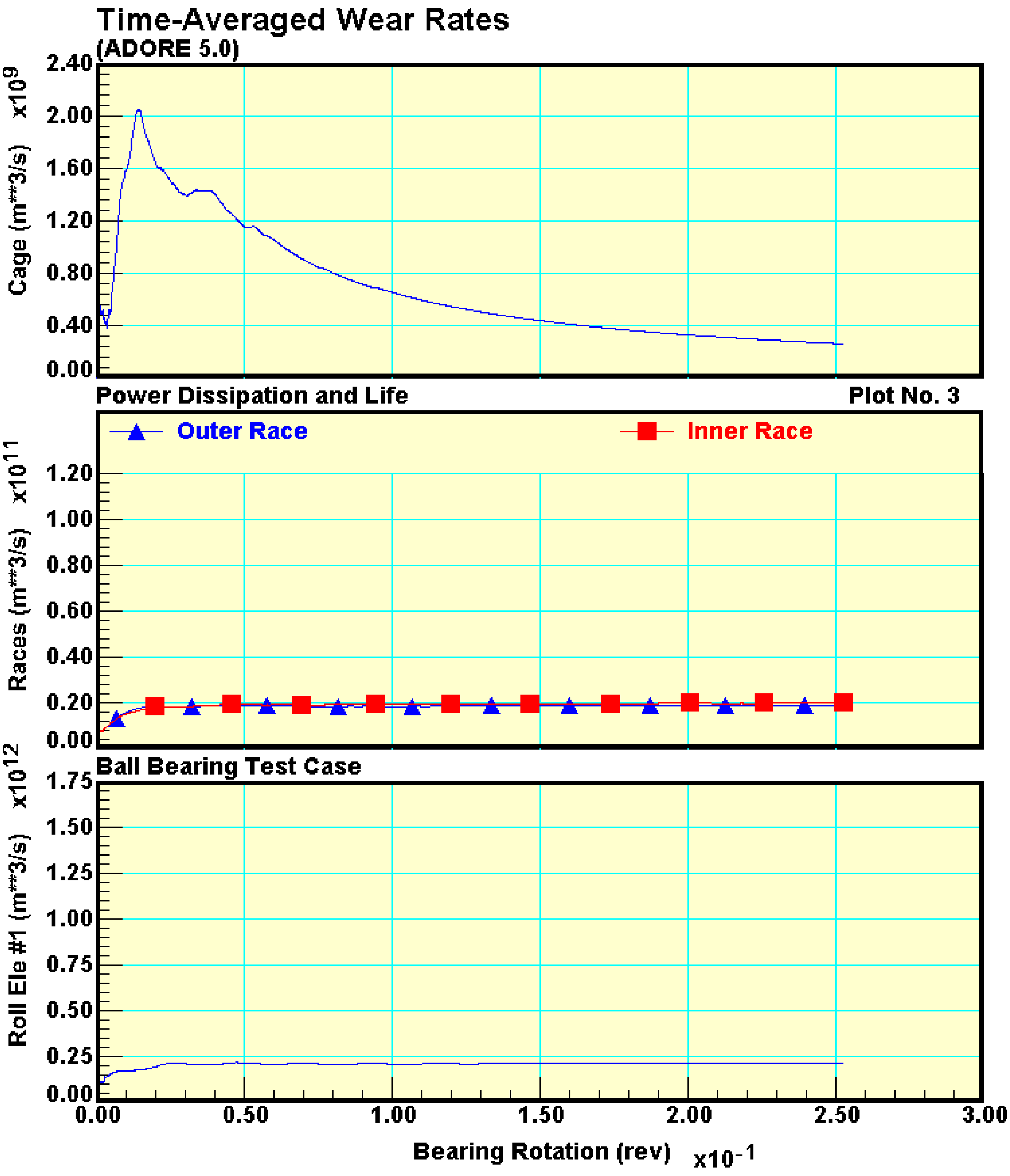

Figure 17. Time-averaged wear rates in the test ball bearing with no thermal interactions.

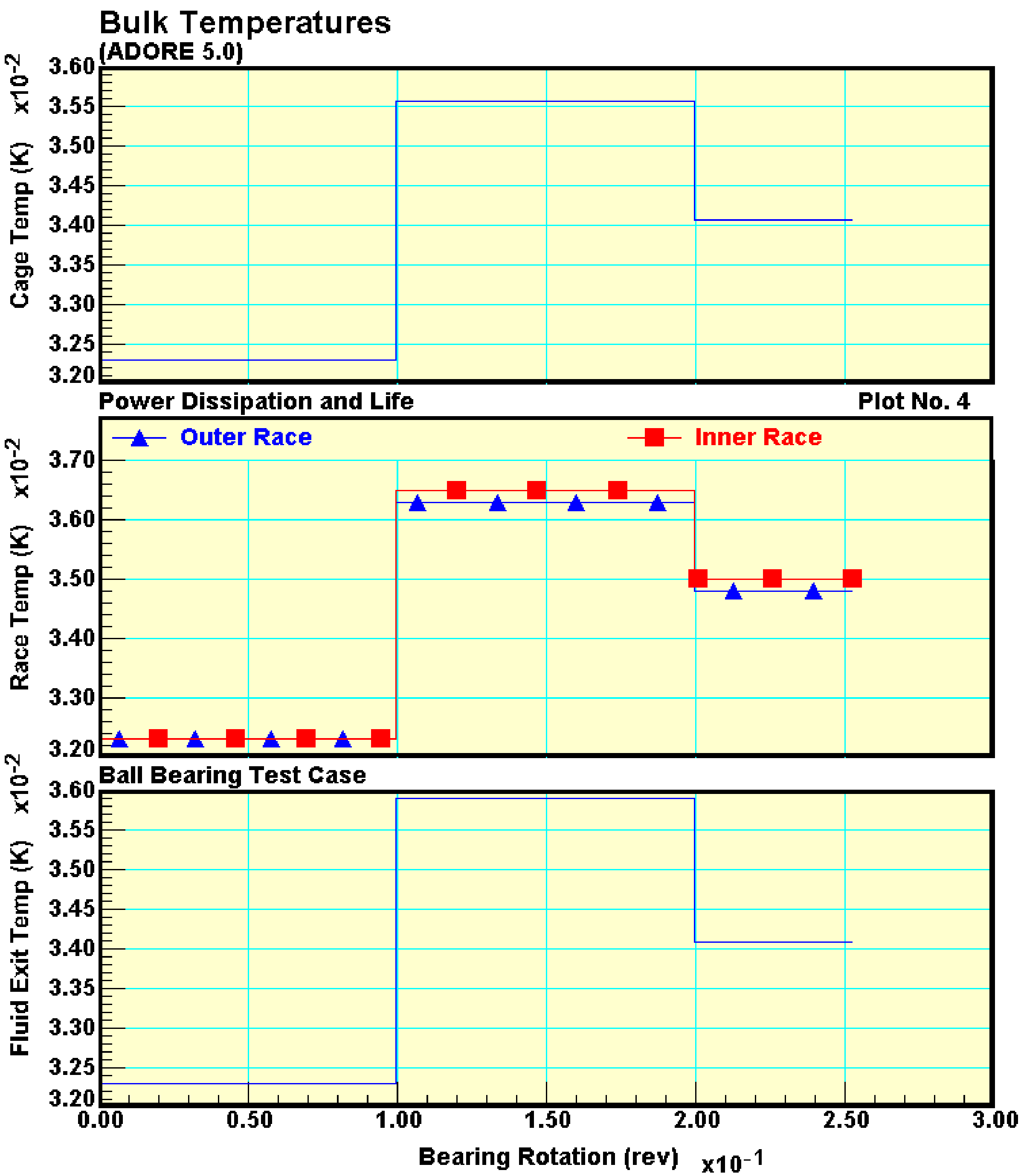

Figure 18. Computed bulk temperatures with no change in bearing geometry.

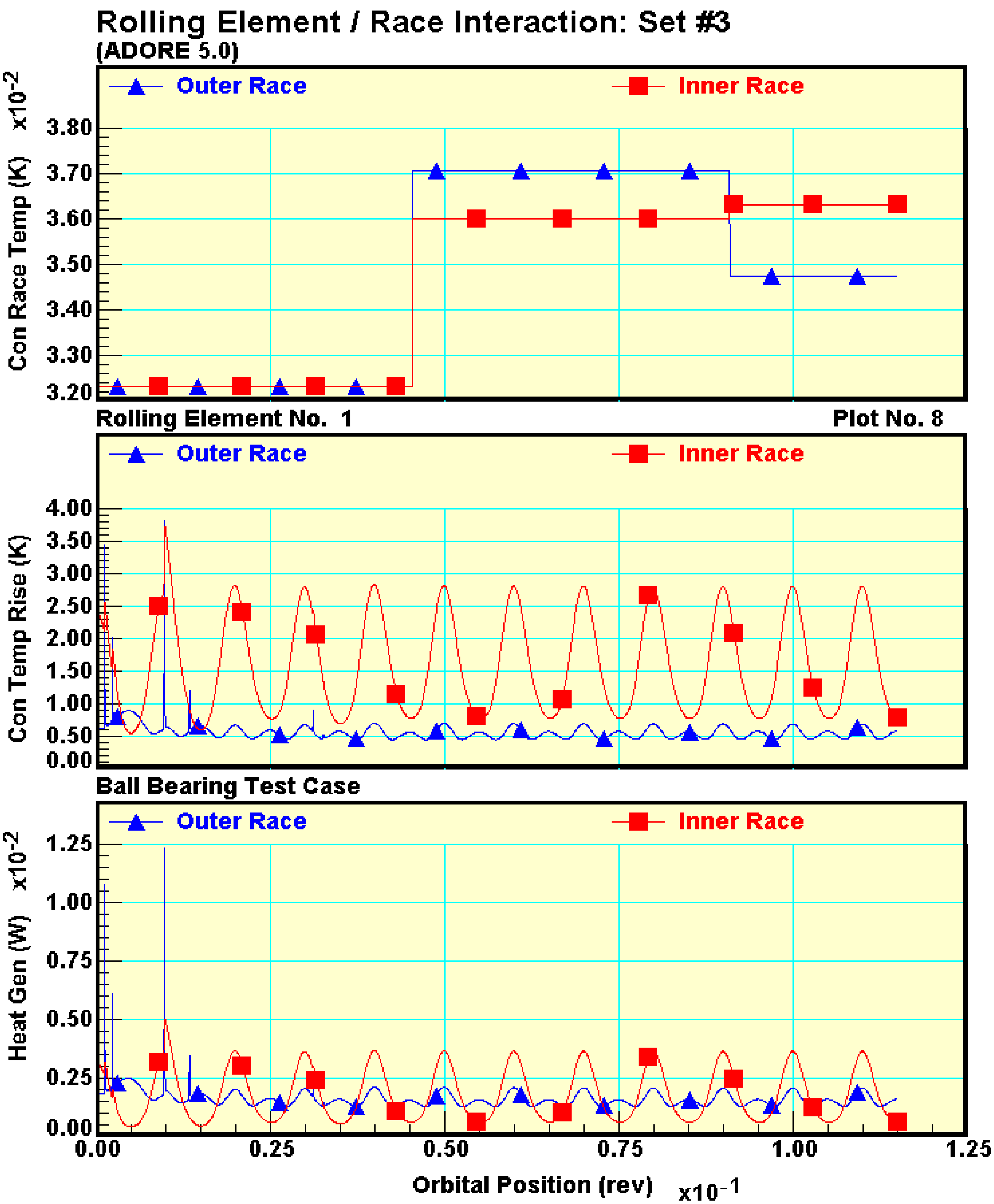

Figure 19. Typical ball/race interaction with no change in internal geometry.

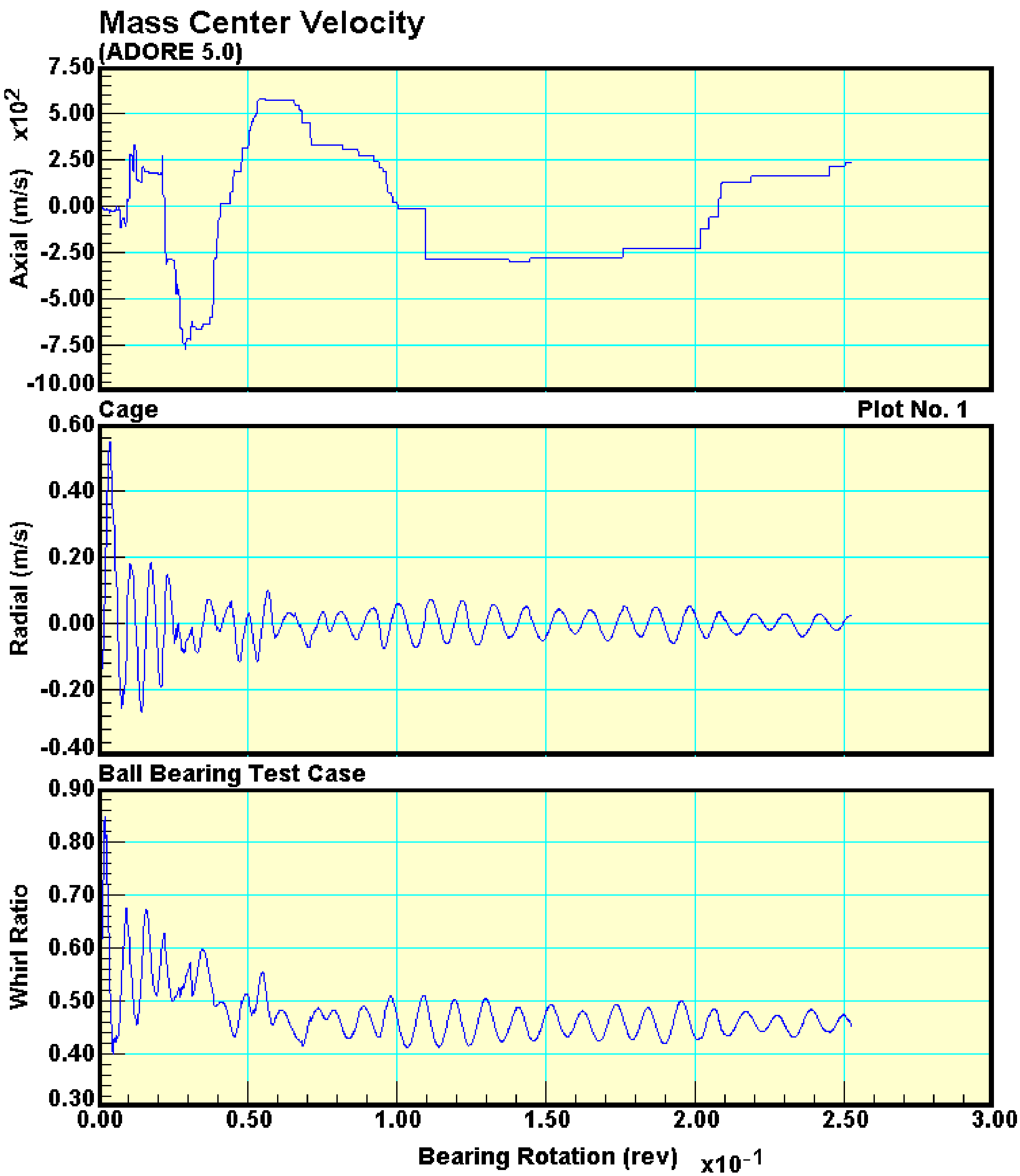

Figure 20. Ball bearing cage mass center velocities in absence of any thermal interactions.

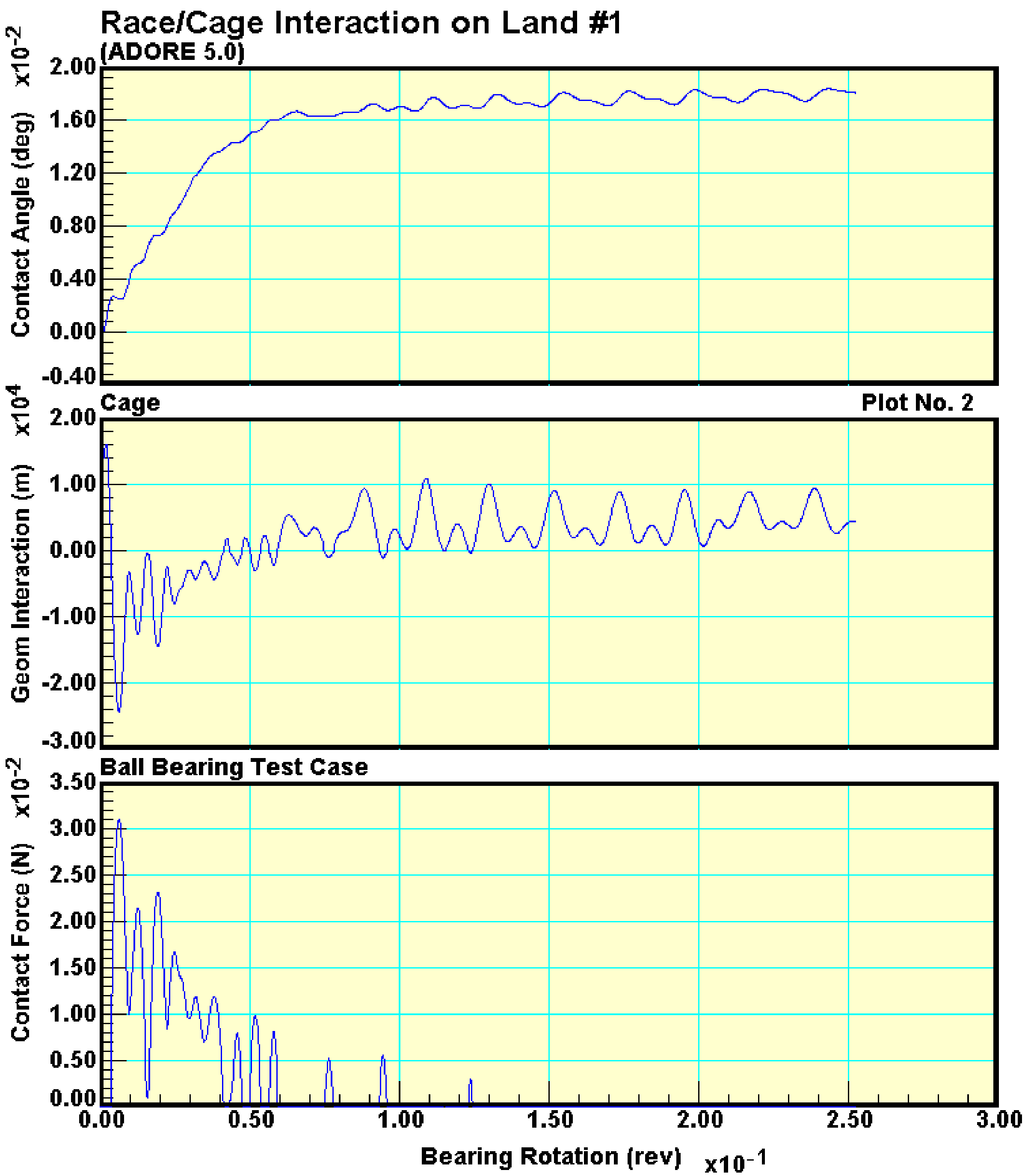

Figure 21. Ball bearing cage/race contacts in absence of any thermal interactions.

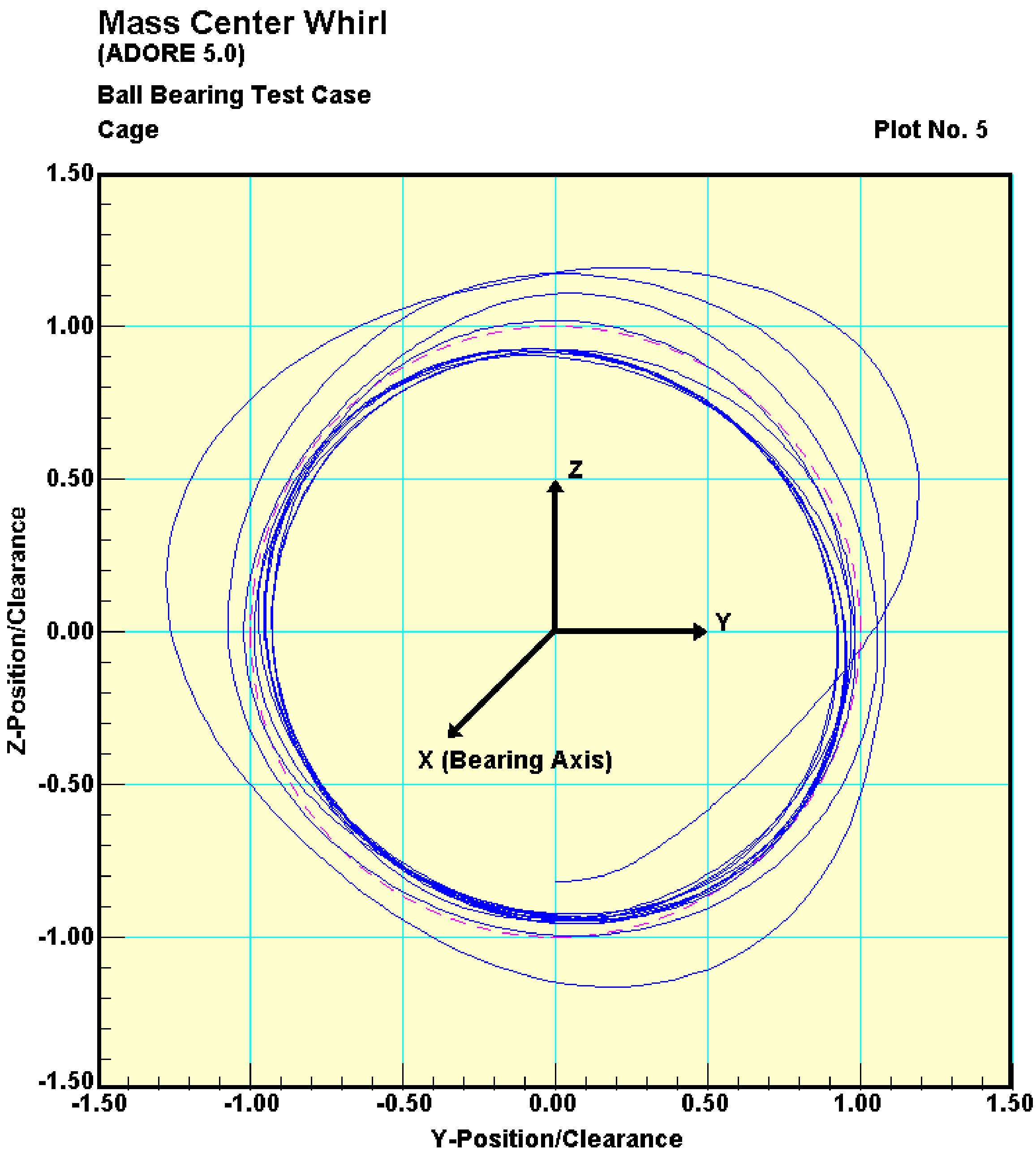

Figure 22. Ball bearing cage mass center whirl orbits in absence of any thermal interactions.

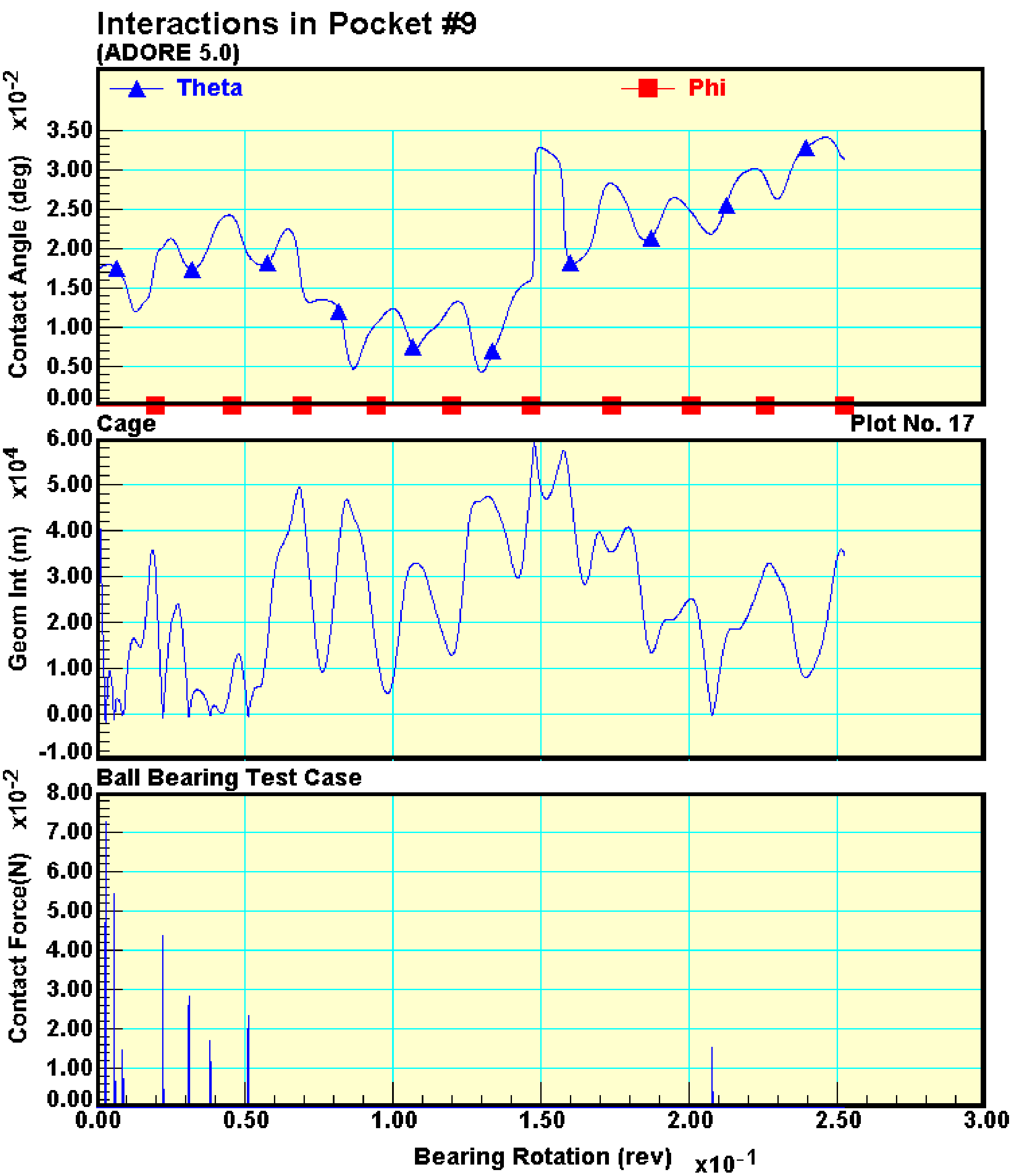

Figure 23. Typical ball/cage contacts in absence of any thermal interactions.

In the second set of results the internal bearing geometry is updated corresponding to the computed bulk temperatures. Also averaging of thermal parameters is now carried out at every shaft revolution. Sudden change in geometry can sometimes lead to a drastic change in bearing interactions. Therefore, considerable care is necessary to handle the mechanical transients when computing the thermal interactions and the resulting change in bearing geometry. Since initially there are fairly high mechanical transients, which are basically a result of the imposed initial conditions, it is quite appropriate to not apply any geometrical changes until some of these transients die out. This is particularly reasonable since the thermal analysis is free of any transients. In the present example, no change in bearing geometry is applied for the first ten shaft revolutions. However the thermal analysis and computation of temperature fields are carried out at each revolution. After the first ten revolutions change in bearing geometry is computed and applied at every shaft revolution. When geometrical changes are applied the bearing internal clearance will change and this may cause a change in applied loads if the races are held at a fixed position. In the example considered here, an equilibrium constraint is applied to move the races and maintain the thrust and radial loads equal to the initially applied values. Thus the applied loads are constant even with the changing bearing geometry.

The overall power loss after accounting for all geometrical changes is basically unchanged as shown in figure 24, and so are the overall mechanical interactions as shown by the time-averaged wear plots in figure 25. The bulk temperatures, after some cyclic variations, stabilize to fairly constant values as seen in figure 26. Local interactions in the ball/race contacts, shown in figure 27, are basically identical to those shown earlier in figure 19. This is primarily due to identical load conditions.

Changes in cage motion due to subtle geometrical changes, are also minimum when the applied loads remain constant. The overall whirl variation, cage/race contacts, whirl orbits and pocket interactions are shown respectively in figure 28, 29, 30 and 31, respectively.

The above results show that when the applied loads are maintained constant, the subtle changes in bearing geometry resulting from thermal interactions do not change the overall bearing behavior. If however, the races are held in a fixed position and the applied loads are permitted to change, there may be a significant change in bearing behavior. The simulations obtained by ADORE will certainly predict such changes in bearing behavior. However, significant experimental validations of such predictions are essential before such predictions may be used in practical design. Such simulations are therefore deferred until some experimental data becomes available.

Another limitation of the current modeling procedures related to the effect of traction as a function of temperature. Although the Newtonian elastohydrodynamic model computes a fairly rigorous temperature and shear stress distribution through the lubricant film, the input rheological constants are derived by regression analysis experimental traction data obtained over a rather limited range of temperature. Thus extrapolation of these model constants over a wider range of temperatures may introduce certain uncertainties. In view of such limitations in the model, the traction model constants are presently not subject to change as the bulk temperatures of the bearing elements vary. However, once reliable data for the traction model coefficients becomes available, elimination of these constraints is straightforward.

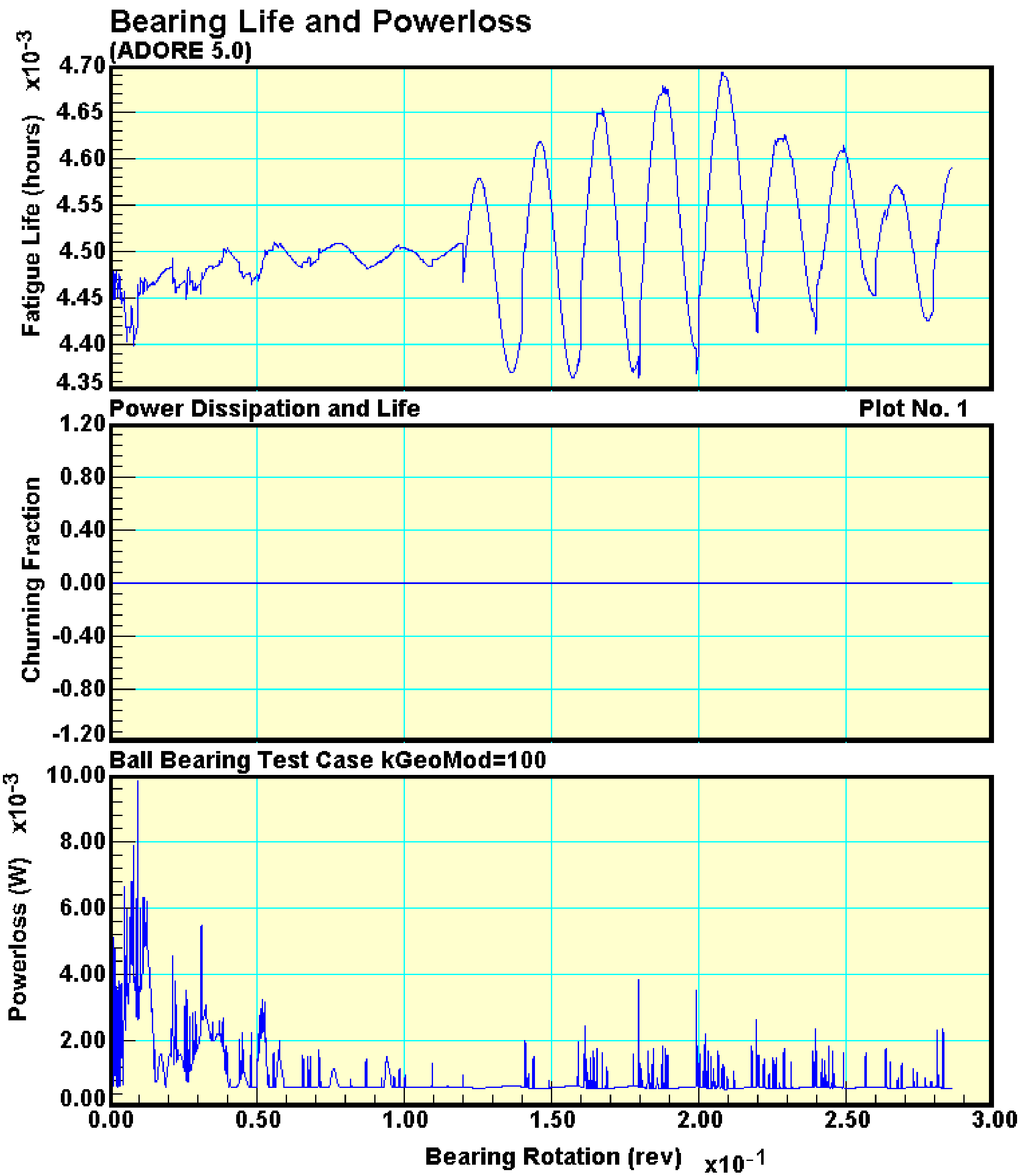

Figure 24. Ball bearing poer loss with time-varying bearing geometry due to thermal effects.

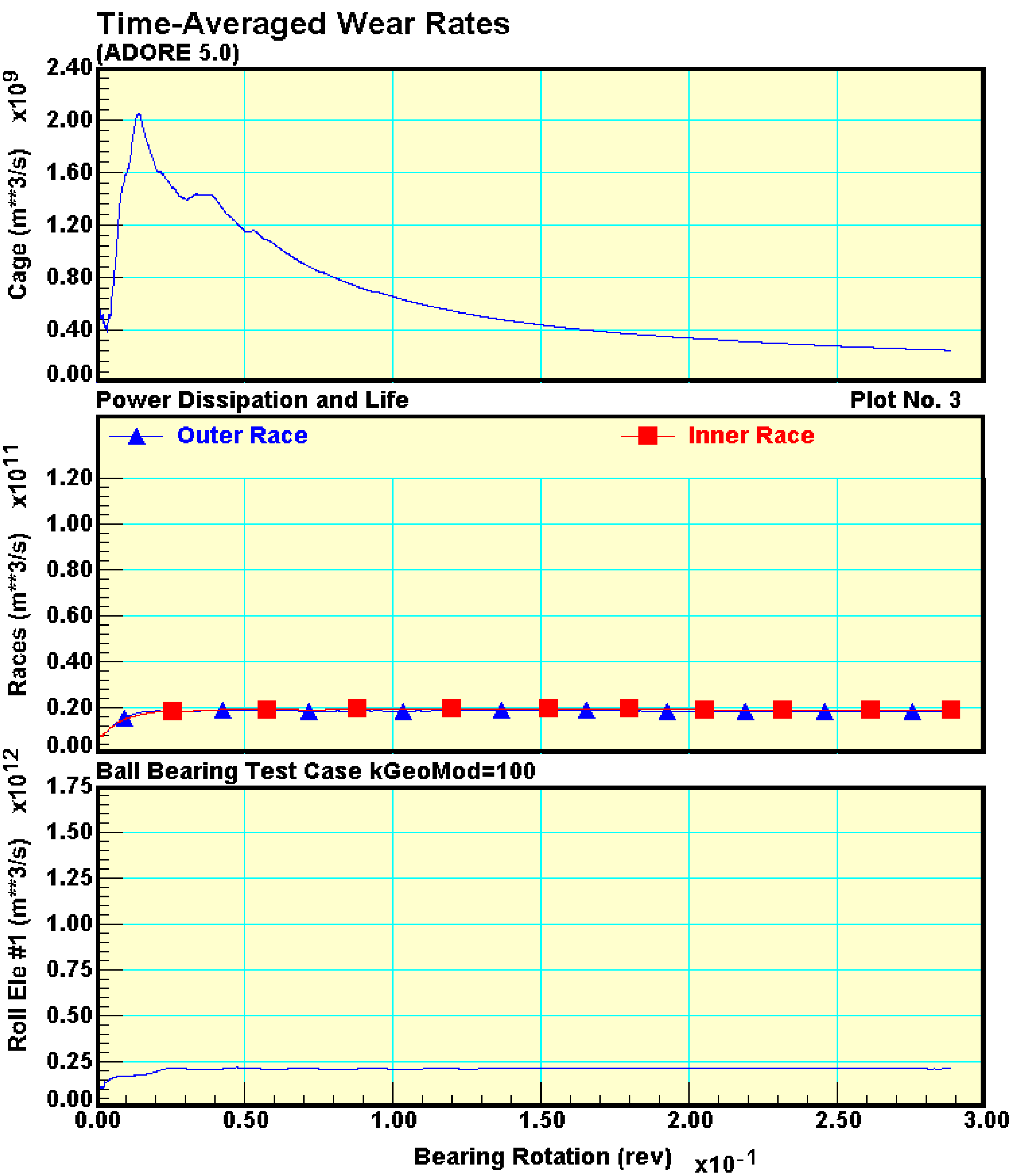

Figure 25. Time-averaged wear rates for the ball bearing with geometrical distortions due to thermal effects.

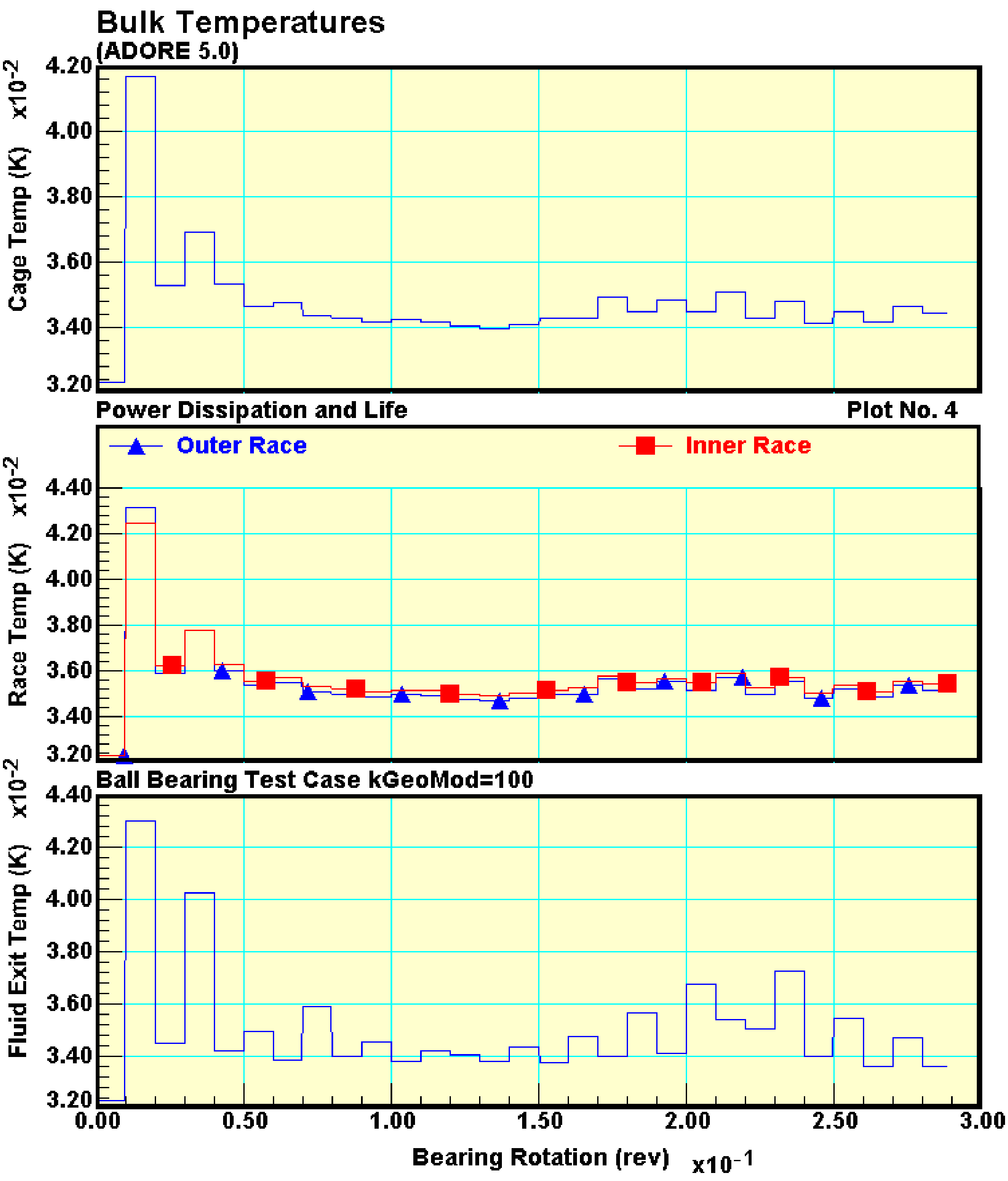

Figure 26. Computed bulk tempeeratues in the ball bearing with thermal effects.

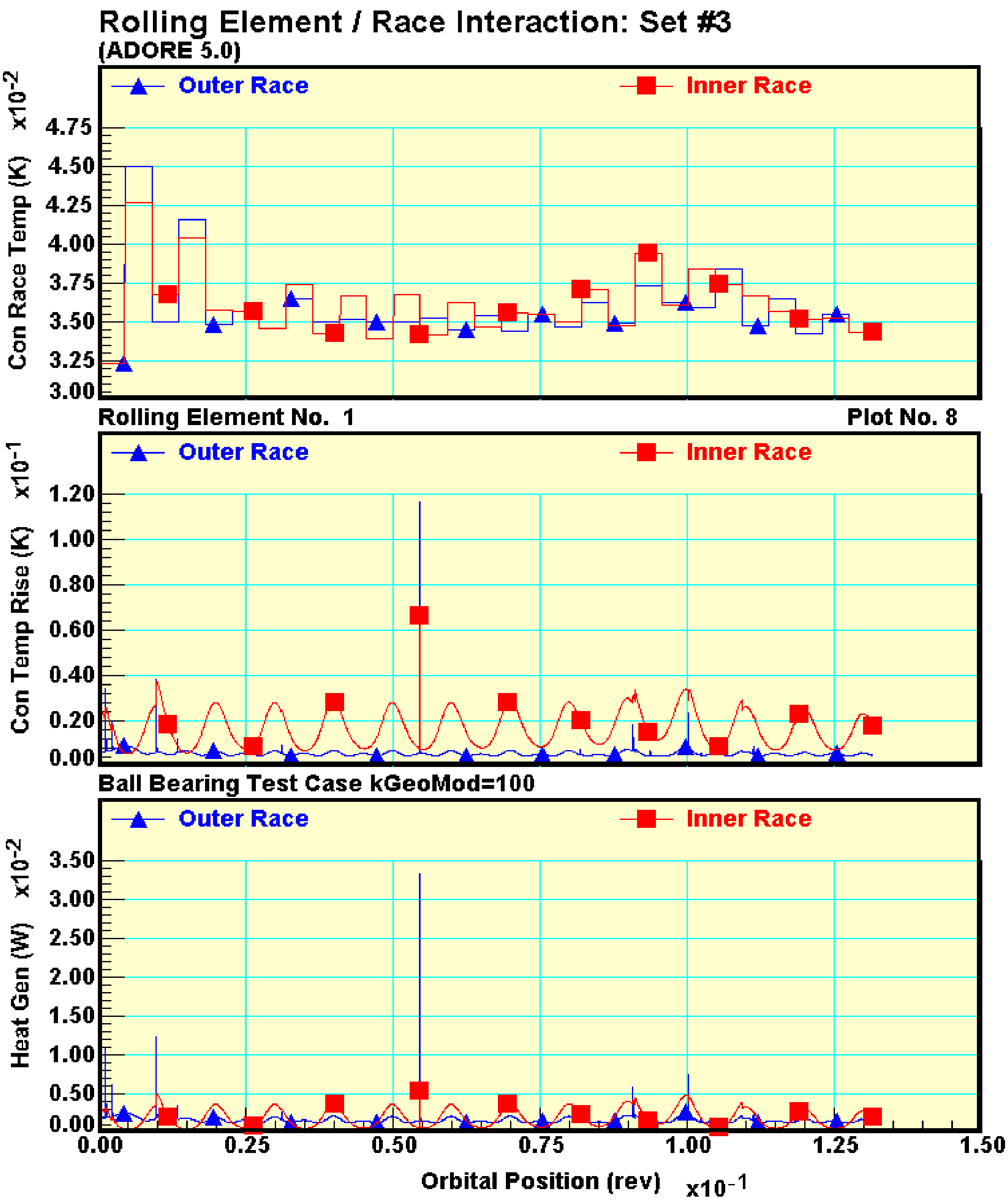

Figure 27. Typical ball/race interactions with thermal effects.

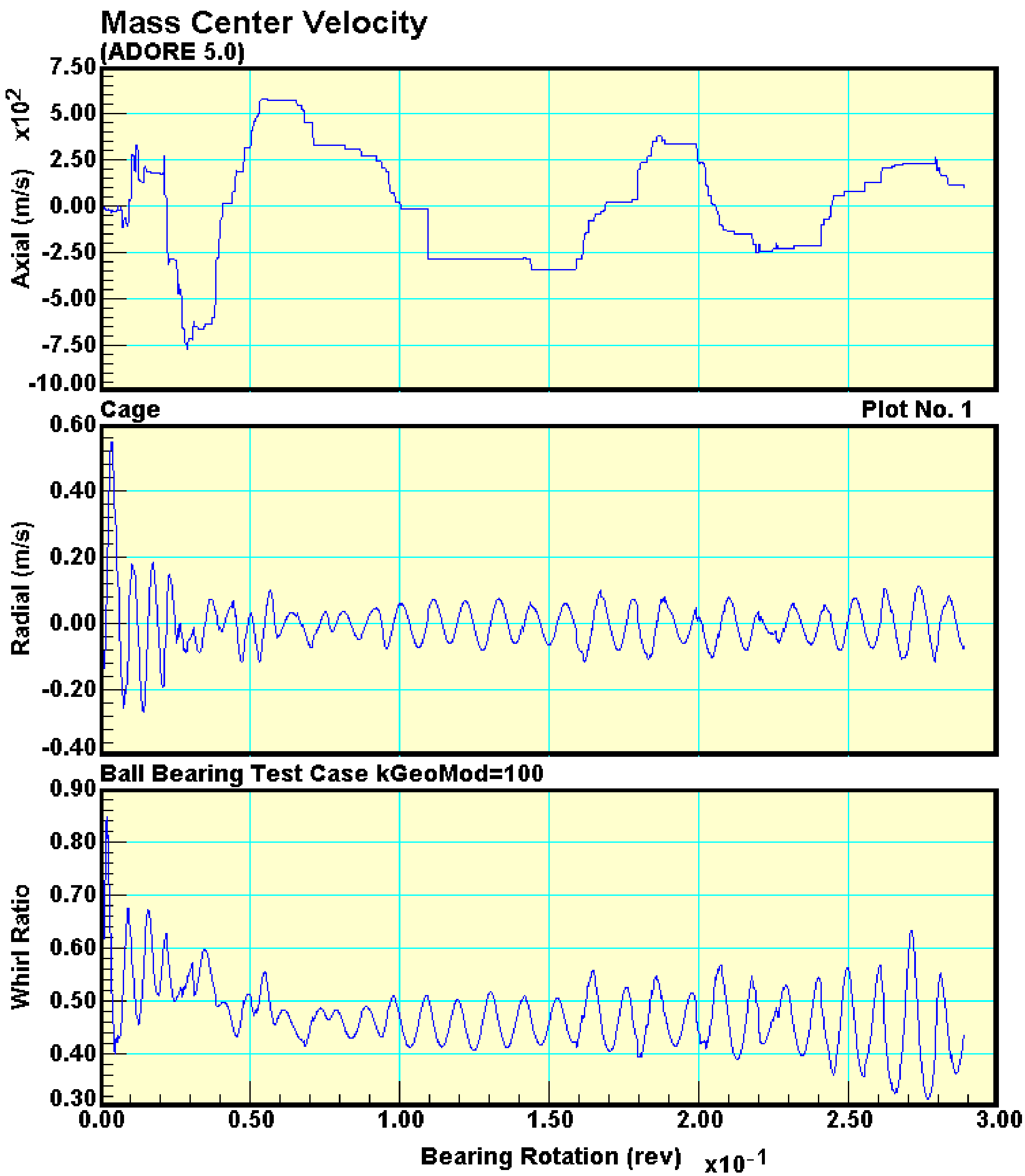

Figure 28. Ball bearing cage mass center velocities with changing bearing geometry as a result of thermal effects.

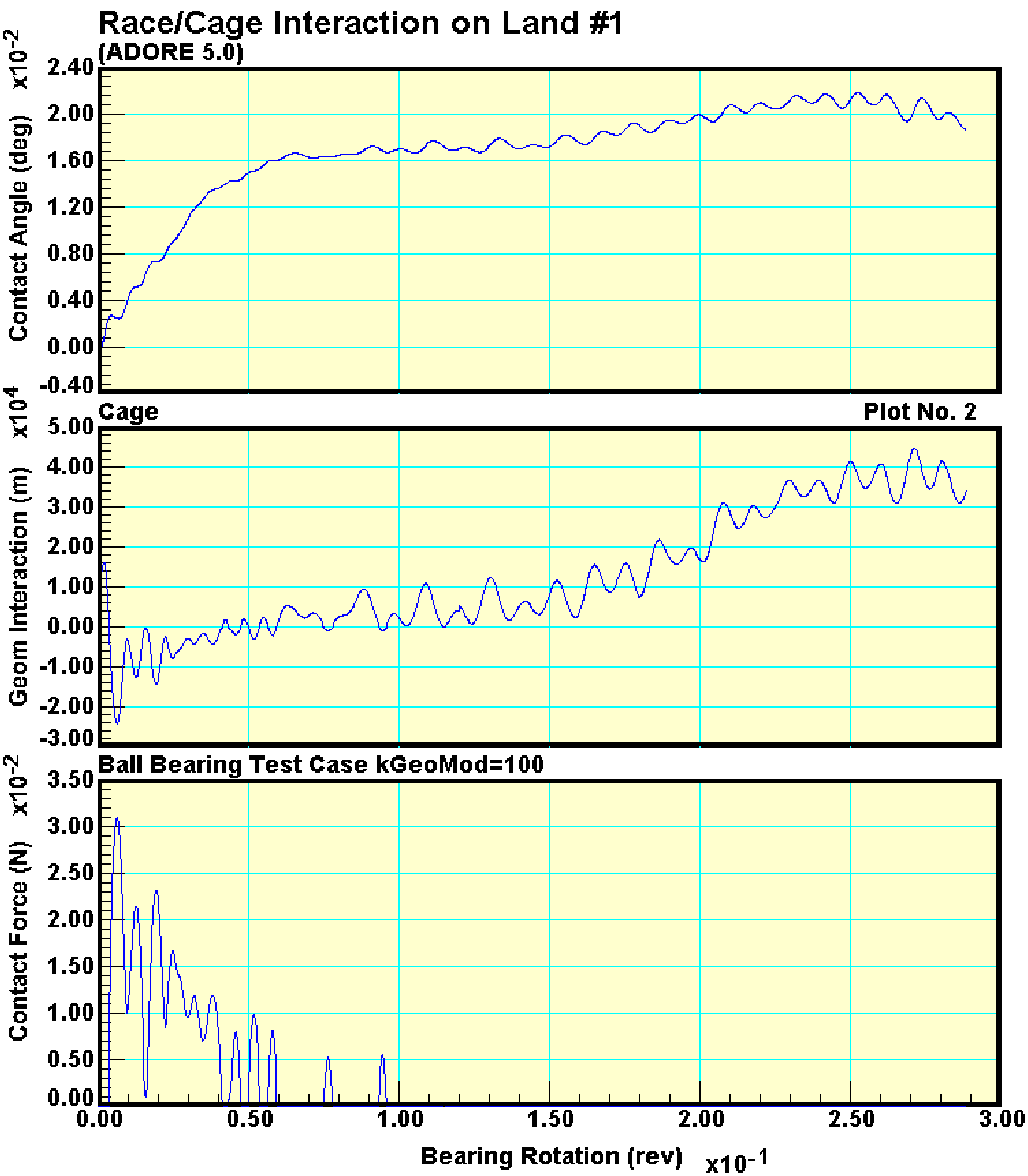

Figure 29. Ball bearing cage/race contacts with thermal effects.

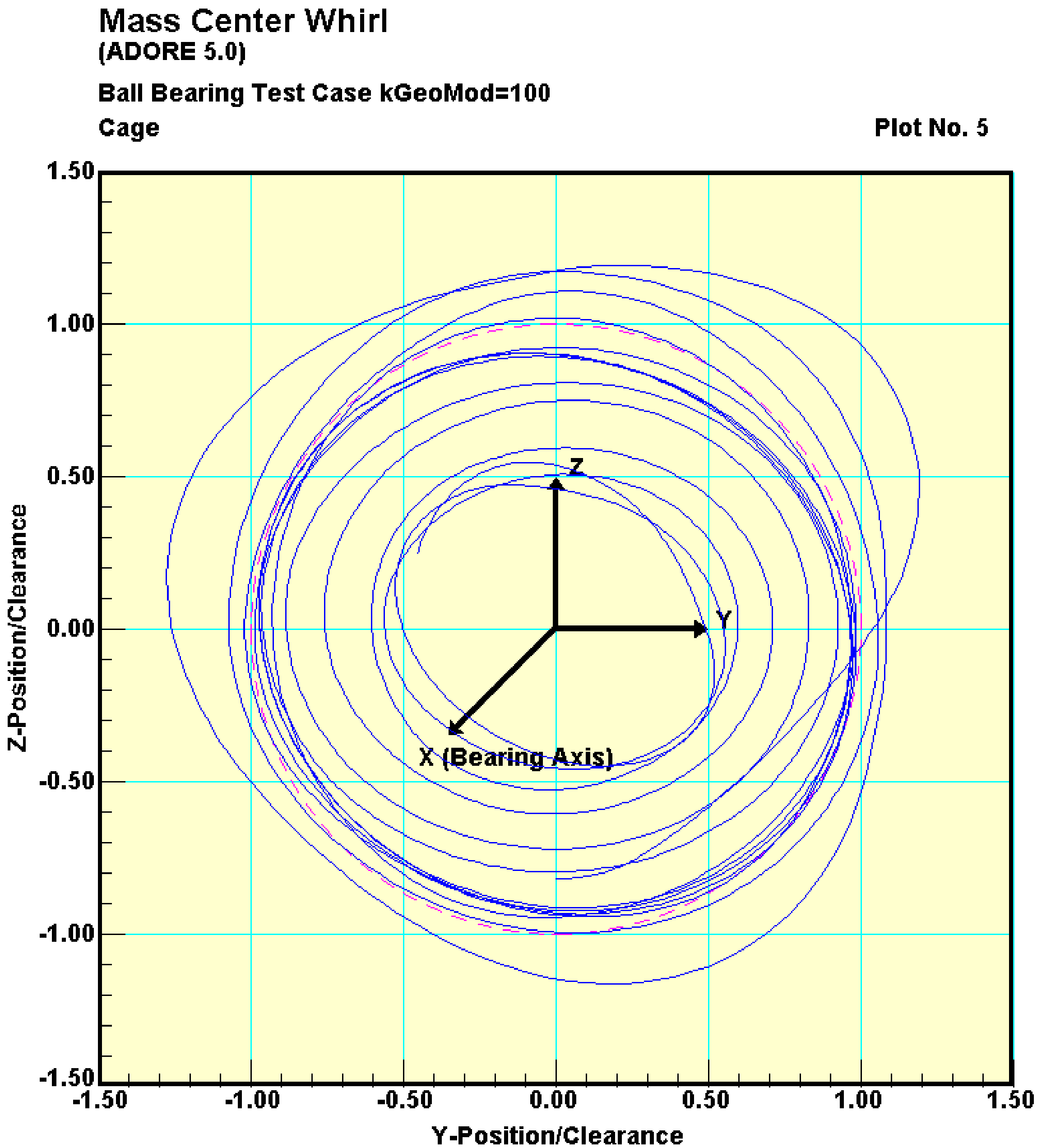

Figure 30. Ball bearing cage mass center whirl orbits with thermal effects.

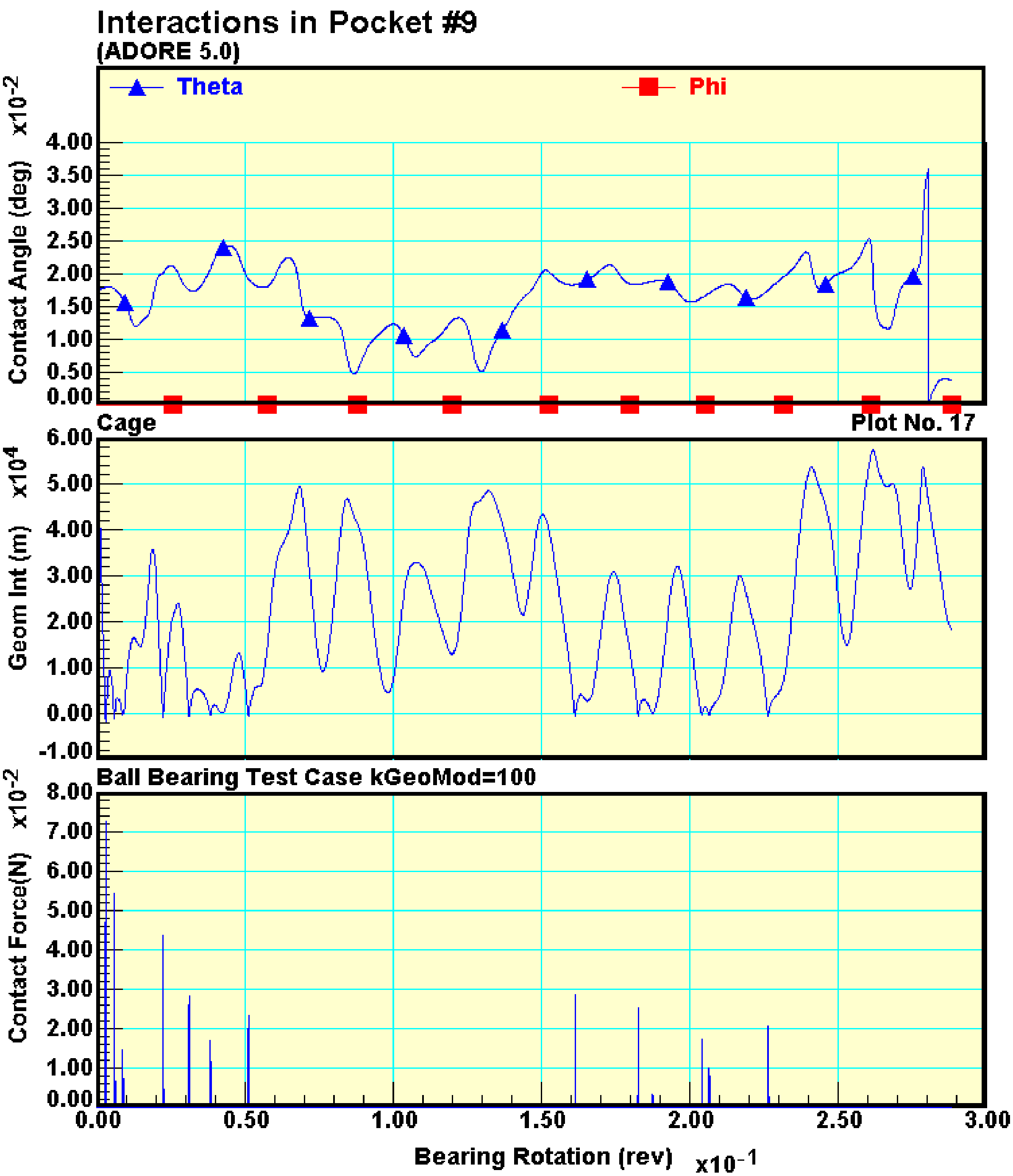

Figure 31. Typical ball/cage interactions with applied thermal effects.

5.4 Test Cylindrical Roller Bearing

A small 30 mm high speed cylindrical roller bearing with ceramic rollers is considered as the test roller bearing. The bearing geometry and operating conditions are described below in table 6.There is no coolant in the bearing, except ambient air.

Table 6: Cylindrical Roller Bearing Geometry and Operating Conditions

Bearing Bore (m)	0.030	Lubricant Type	Arbitrary
Bearing Outside Diameter (m)	0.055	Traction Model	Figure 5-19
Number of Rollers	8	Cage Friction Coefficient	0.050
Roller Diameter (m)	0.0080	Applied Radial Load (N)	1,000
Roller Length (m)	0.0080	Shaft Speed (RPM)	70,000
Central Land Width (m)	0.0020	Room Temperature (K)	294
Crown Radius (m)	0.250	Outer Race Fit (m)	0.
Corner Radius	0.000250	Inner Race Fit (m)	0.000020
Internal Diametral Clearance (m)	0.000040		
Pitch Diameter (m)	0.0430		
Cage Outer Diameter (m)	0.0450	Coolant Type	Ambient Air
Cage Inner Diameter (m)	0.040		
Cage/Race Guidance Type	Outer		
Cage Width (m)	0.010		
Cage Guide Land Clearance (m)	0.00010		
Cage Guide Land Width (m)	0.0010		
Cage Pocket Clearance (m)	0.00010		

An arbitrary traction model, simulating solid lubrication is assumed, with relatively high traction coefficients. ADORE option for an arbitrary traction slip relation is used to prescribe the traction model. The relationship prescribed in shown in figure 32. Detailed input data to ADORE and typical print output corresponding to the present bearing are summarized in Appendix D.

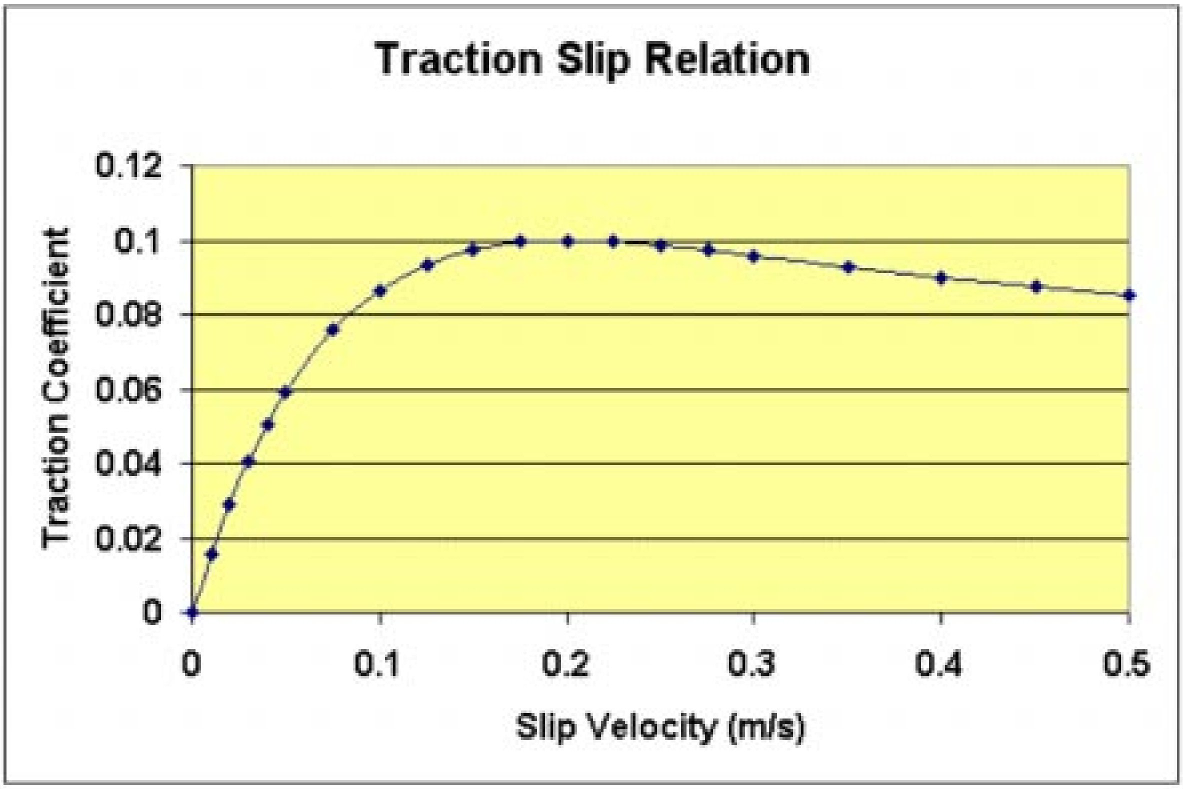

Figure 32. Traction-Slip relation for the test roller bearing.

5.5 Thermal Interactions Cylindrical Roller Bearing

Thermal modeling, once again, is carried out in two parts. First the heat generation is averaged over a shaft revolution and no change in bearing geometry is assumed. Typical results are shown in figures 33 to 38.

The overall heat generation in this bearing under the prescribed operating conditions is quite low as shown in figure 33. The peaks in the curve are a result of roller/cage contacts while the small humps are due to cage/race contacts. As may be expected, he temperature change under such conditions is small, as seen in figure 34. The small step change in bulk temperatures at every shaft revolution is a result of the thermal analysis. Note that there is no circulating coolant in this bearing except for ambient air. The fluid exit temperature, therefore represents the ambient air temperature near the rolling elements. Details of the local thermal interactions at the roller/race contacts are shown in figure 35. Again the temperature rise in the contact and the change in race temperatures at the contact are small due to low value of heat generation.

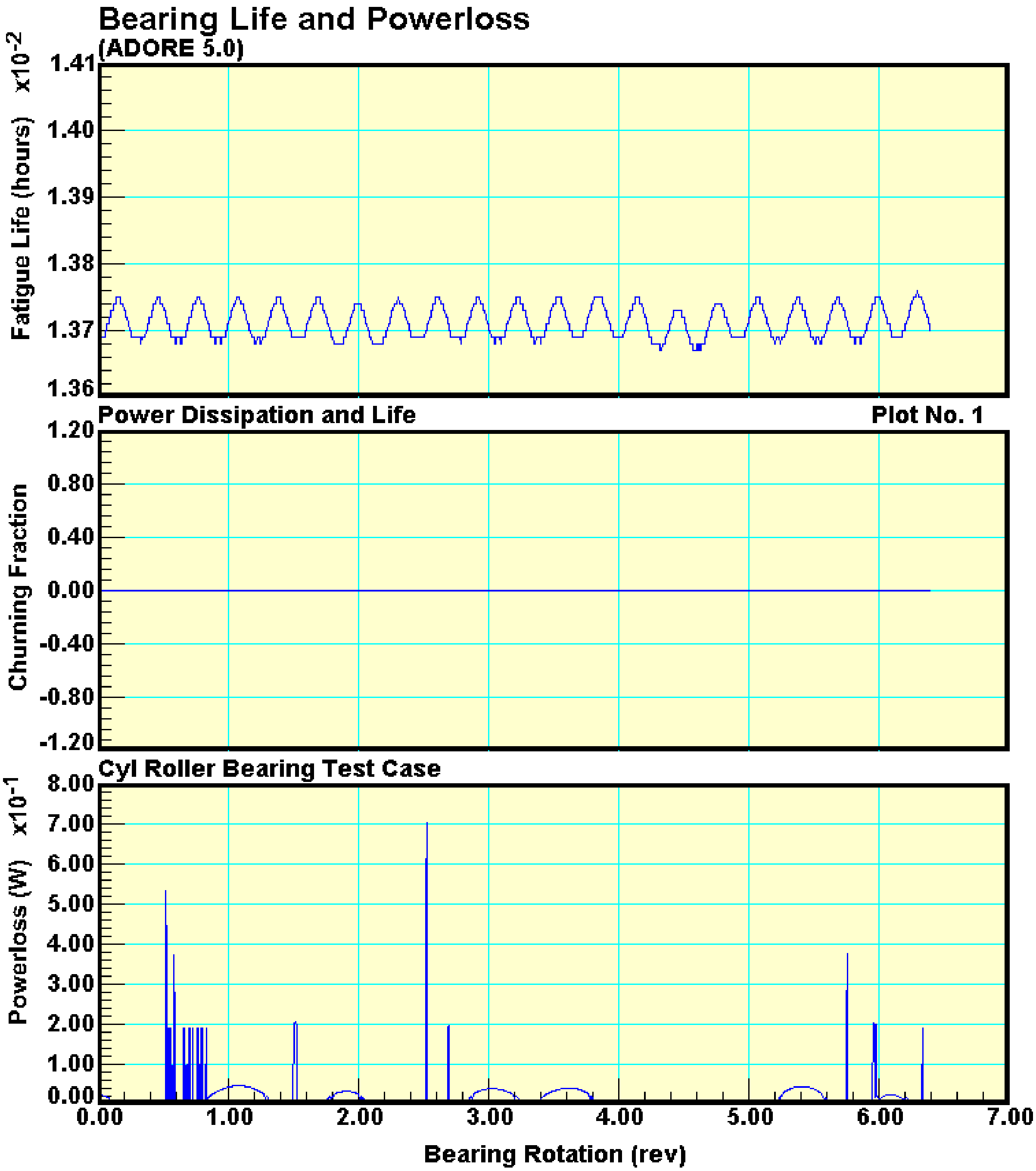

Figure 33. Roller bearing powerloss with no thermal effects.

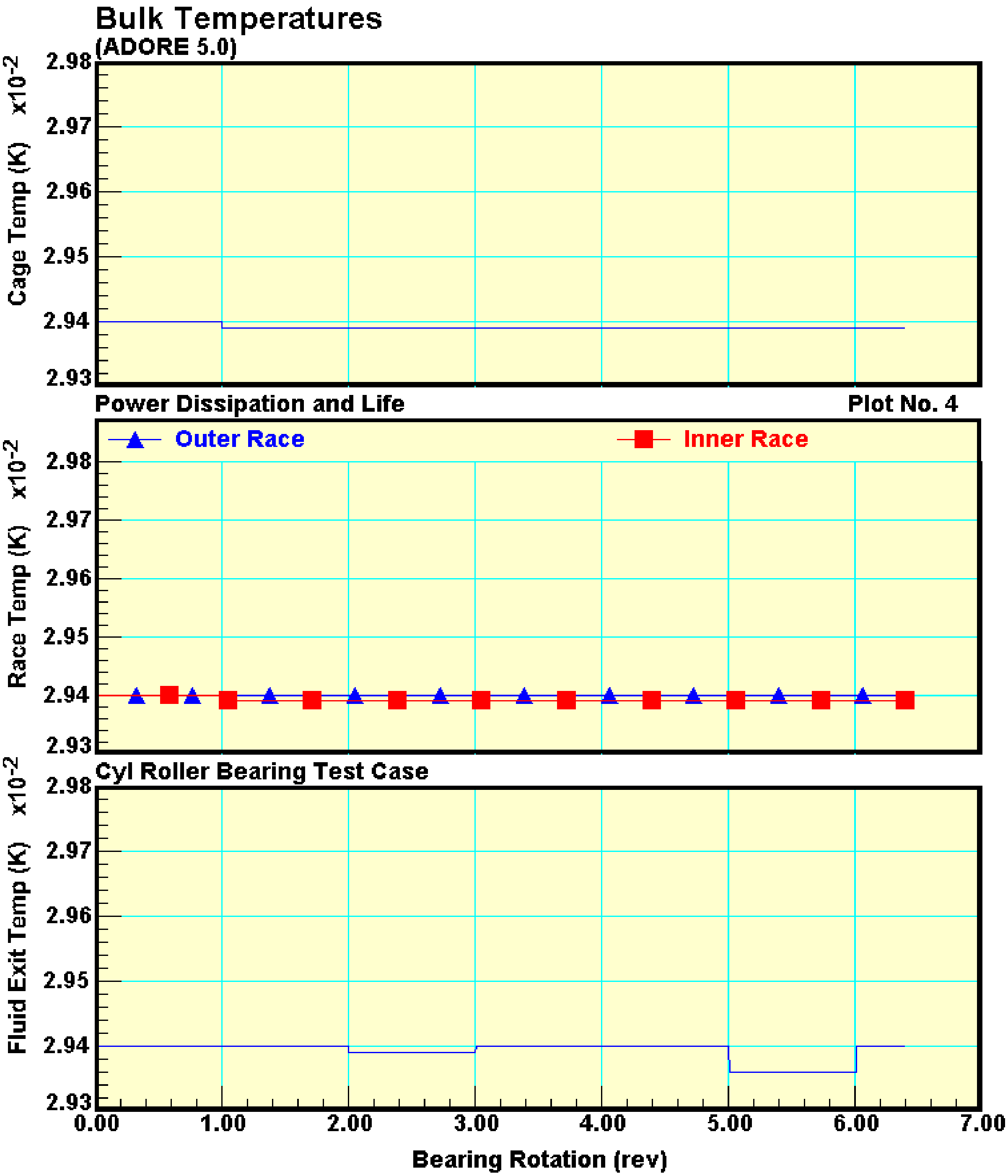

Figure 34. Computed bulk temperatures in the roller bearing with no thermal effects.

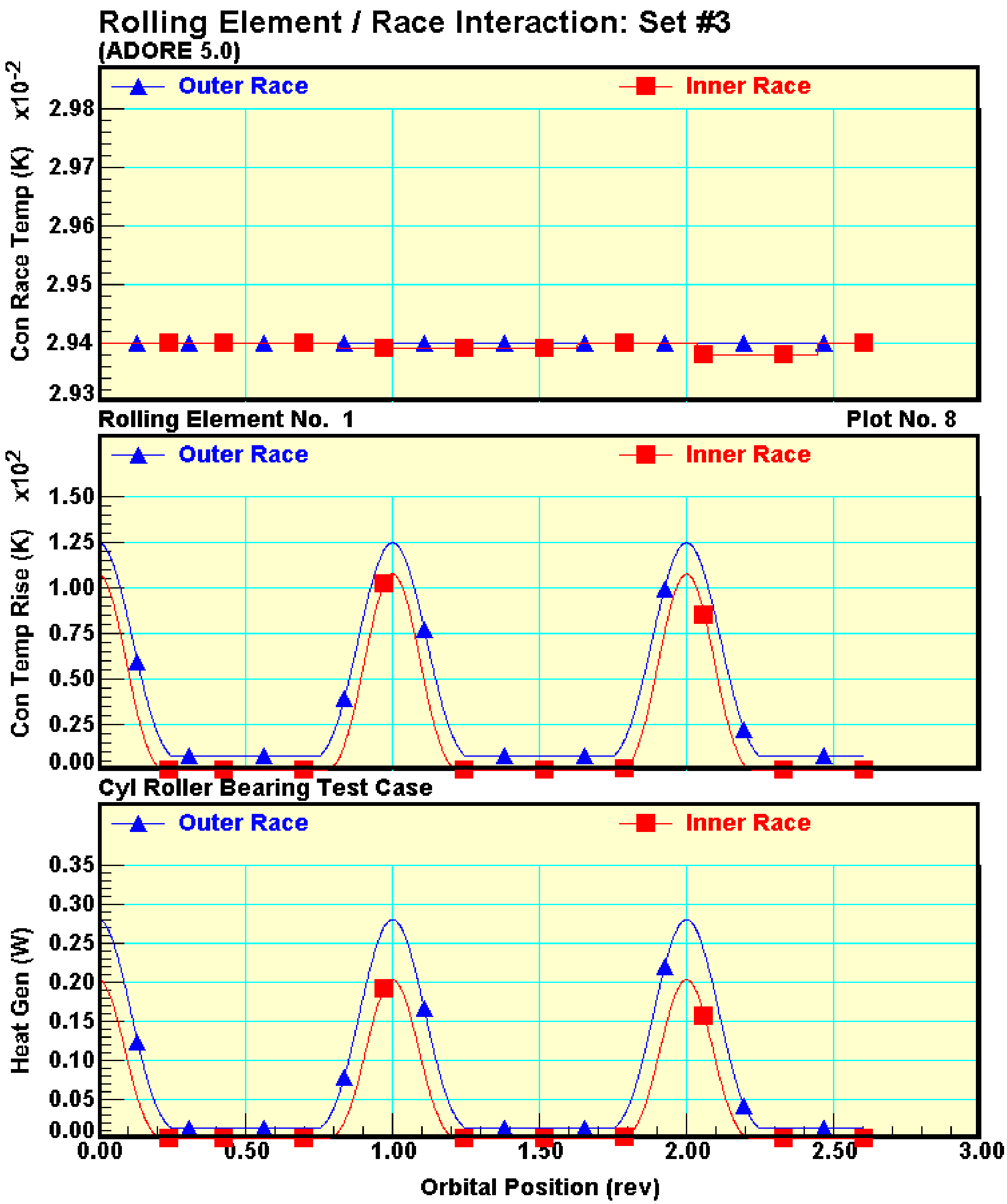

Figure 35. Typical roller/race interactions with no thermal effects.

Cage interactions are also minimum. Mass center whirl is practically unchanged from the initially prescribed circular motion. Small variations in whirl velocity, the cage/race forces and the whirl orbits are shown in figures 36, 37 and 38 respectively.

In the next simulation, internal bearing geometry is updated corresponding to the computed temperature field. Again the heat generation is averaged over every shaft revolution and after computing the temperatures the bearing geometry is updated. Again at each instant of altered temperatures a quasi-static constraint is applied to maintain the applied radial load at the initially prescribed value. Primarily due to rather low value of overall heat generation, the roller/race contacts are practically unchanged. The overall power loss, computed bulk temperatures and typical roller/race interactions are shown respectively in figures 39, 40 and 41.

Cage motion resulting from the modified cage/race clearance with varying temperatures is perhaps more significant. The whirl velocity variation is slightly larger, as seen in figure 42. This is probably due to the increase cage/race interaction, as shown by the guide land force variation plotted in figure 43. Mass center whirl still remains circular, as seen in figure 44, although there is a small shift from the initial circular orbit.

Again it must be remembered that the predicted changes in bearing performance as a function of changing geometry due to thermal effects must be validated experimentally before attaching a significant design significance to them. In addition, the roller/race traction relation may also vary significantly with changing race and/or roller temperatures. Again, these models are inputs to ADORE and significant experimental development is necessary before the thermal modeling procedures, as implemented in ADORE, may be effectively used for practical bearing design.

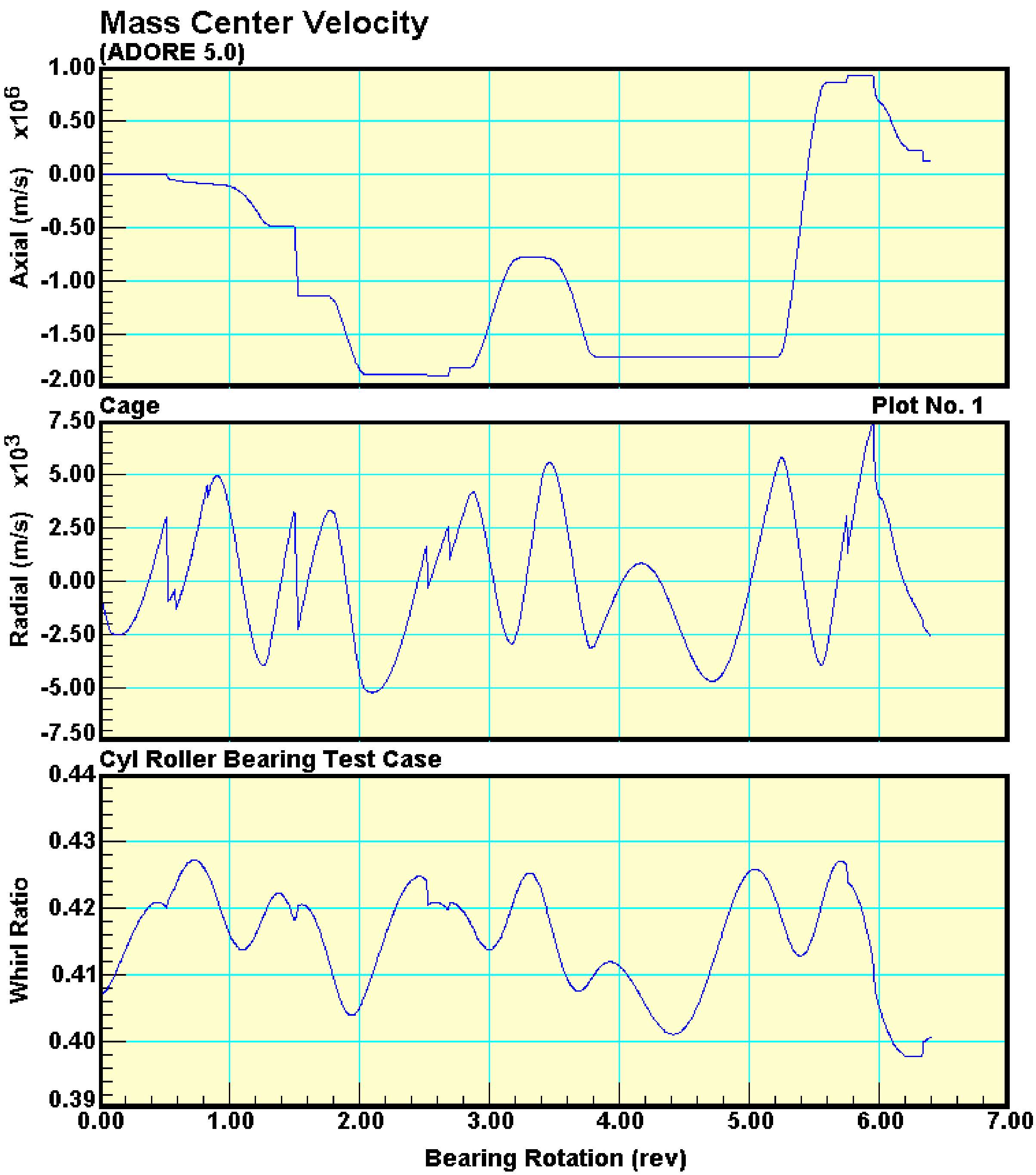

Figure 36. Roller bearing cage mass center velocities with no thermal effects.

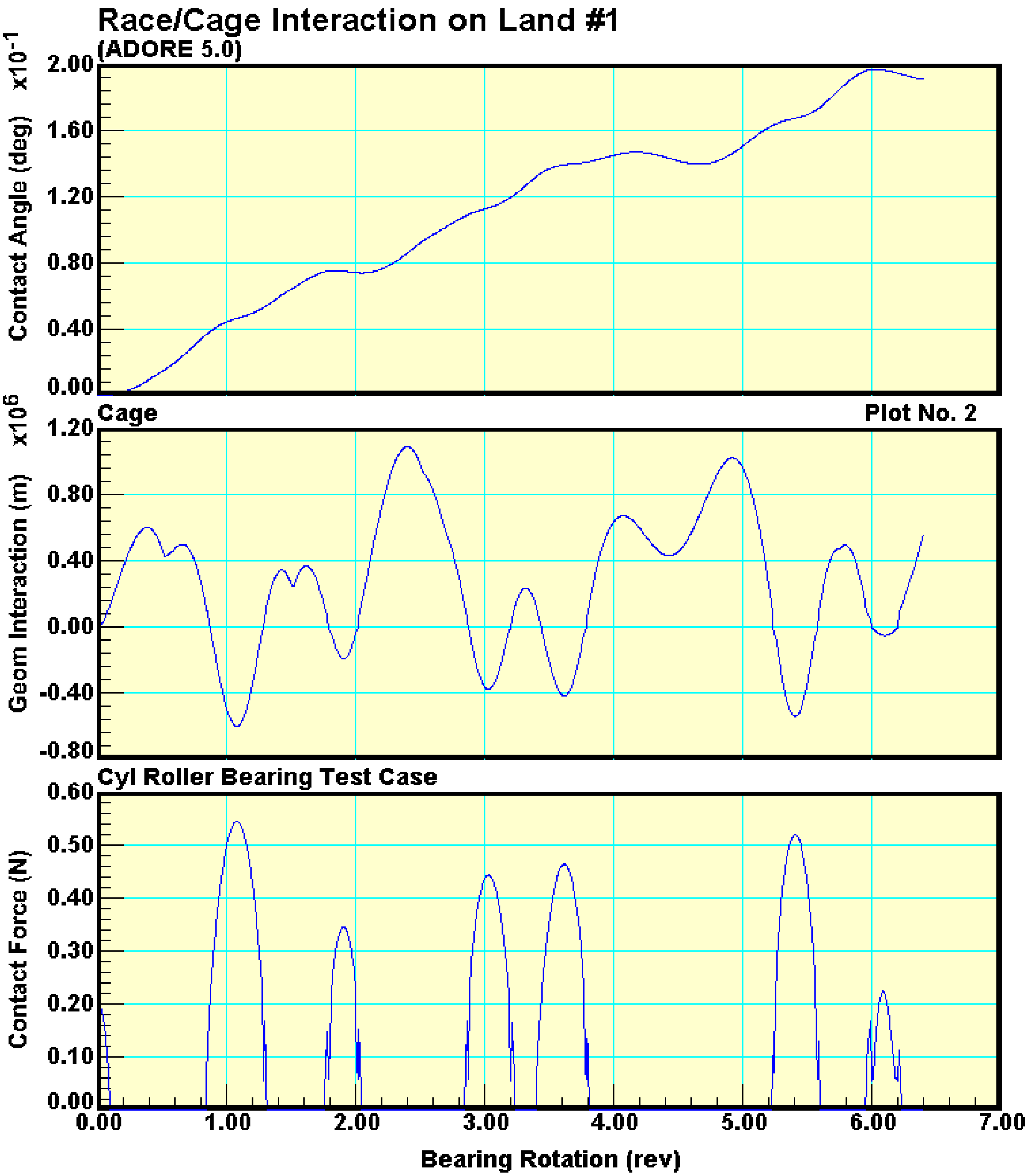

Figure 37. Cage/race contacts in the roller bearing with no thaermal effects.

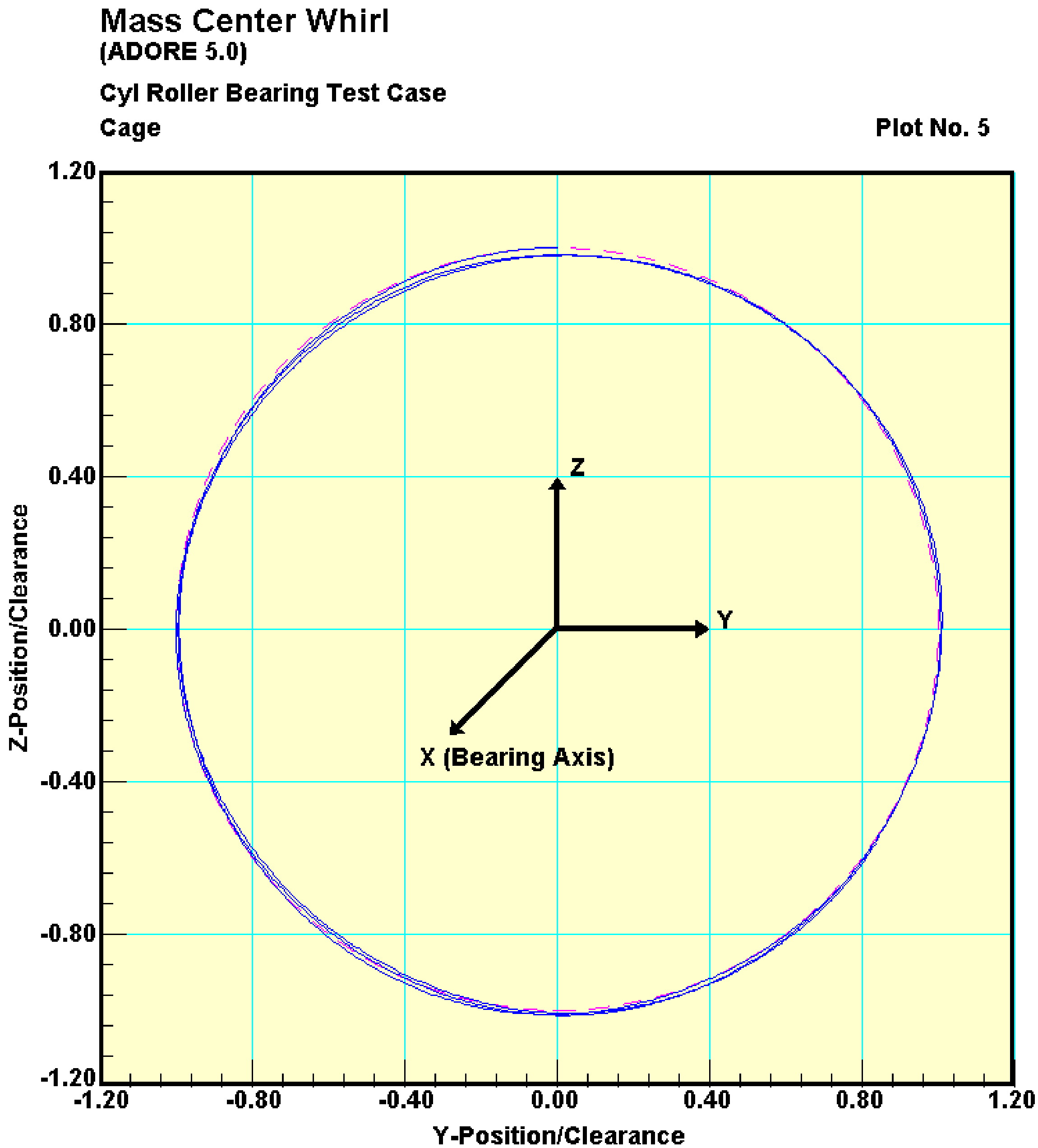

Figure 38. Roller bearing cage whirl orbits with no thermal effects.

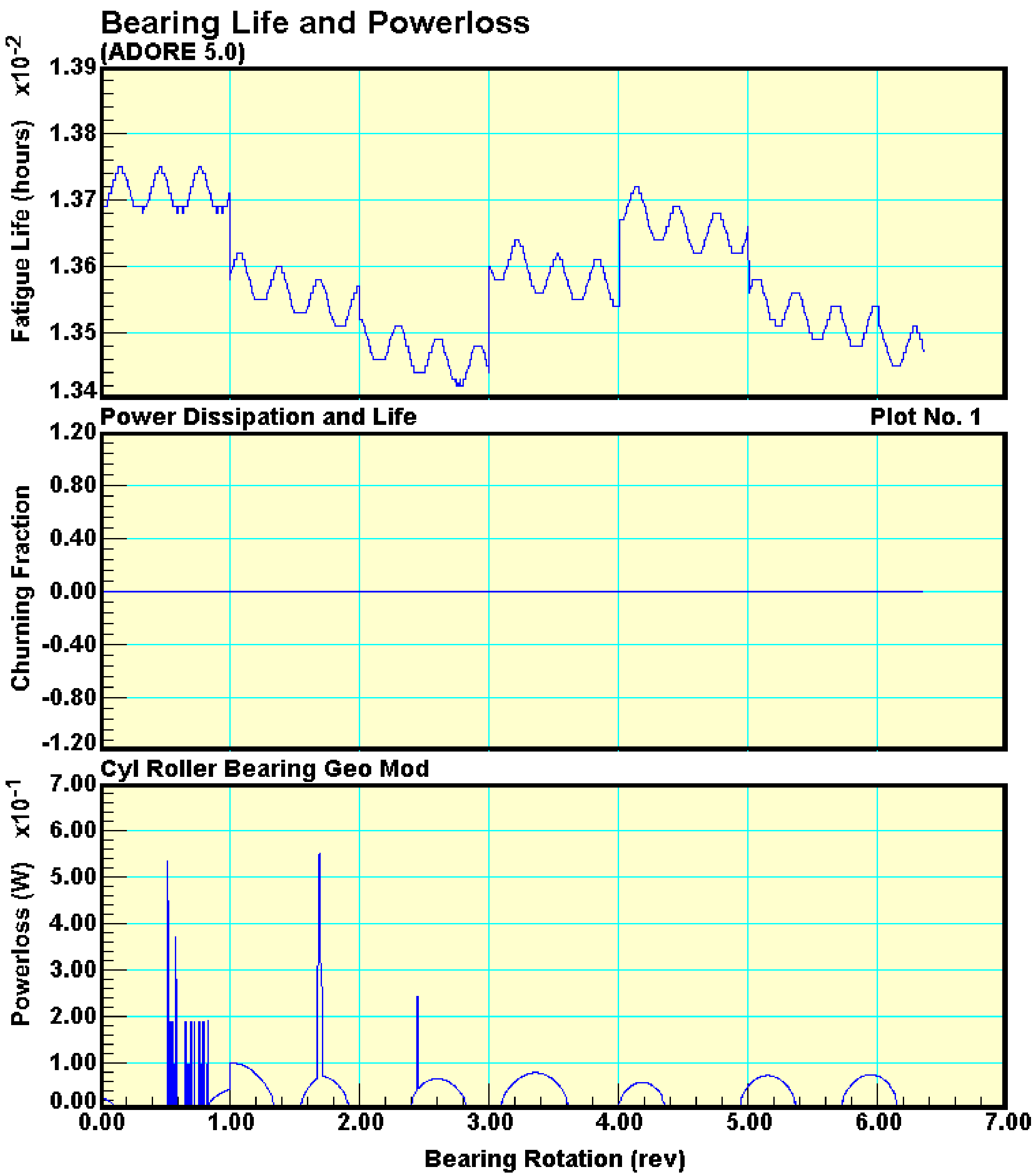

Figure 39. Roller bearing powerloss with all thermal effects.

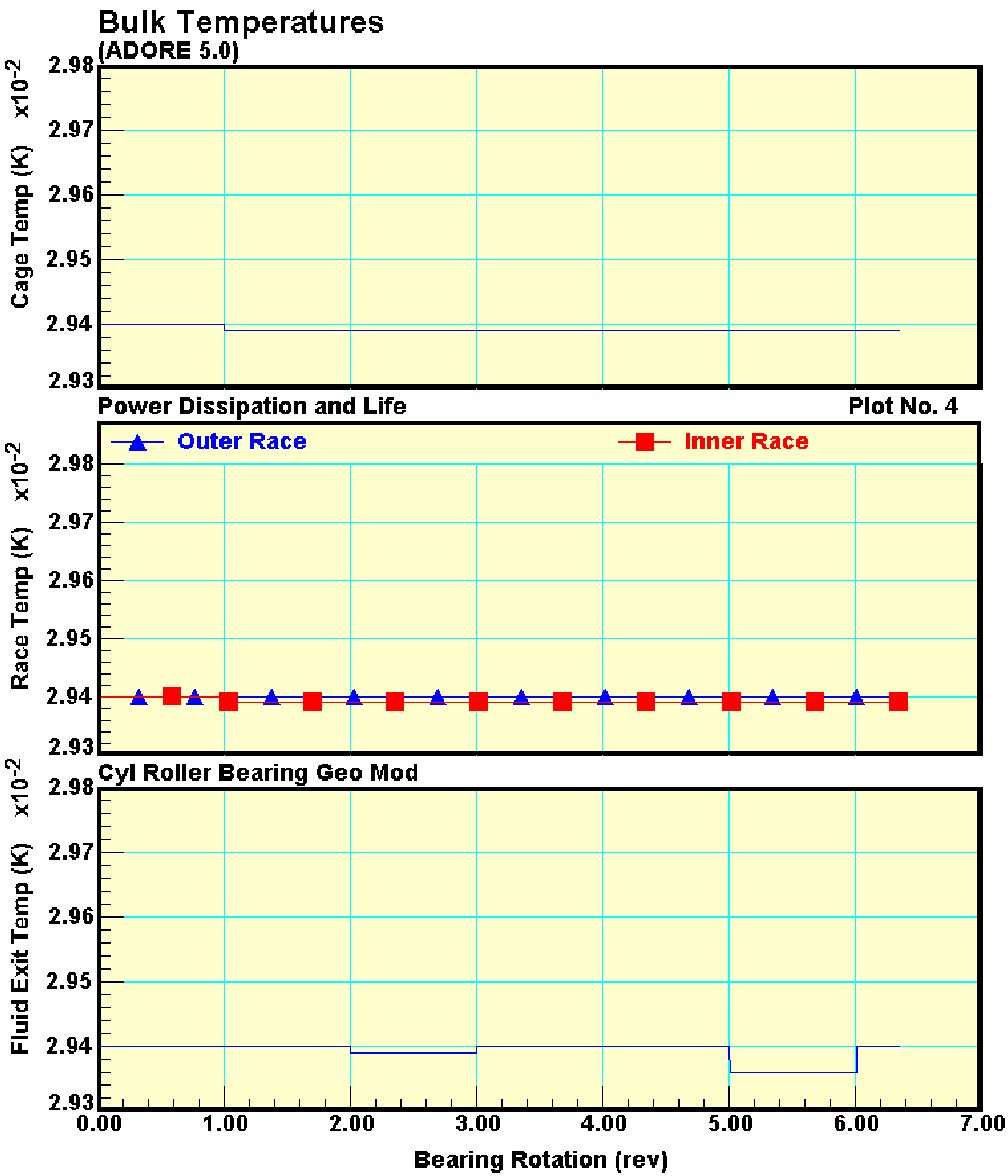

Figure 40. Computed roller bearing bulk temperatures with changing bearing geometry as a rsult of thermal effects.

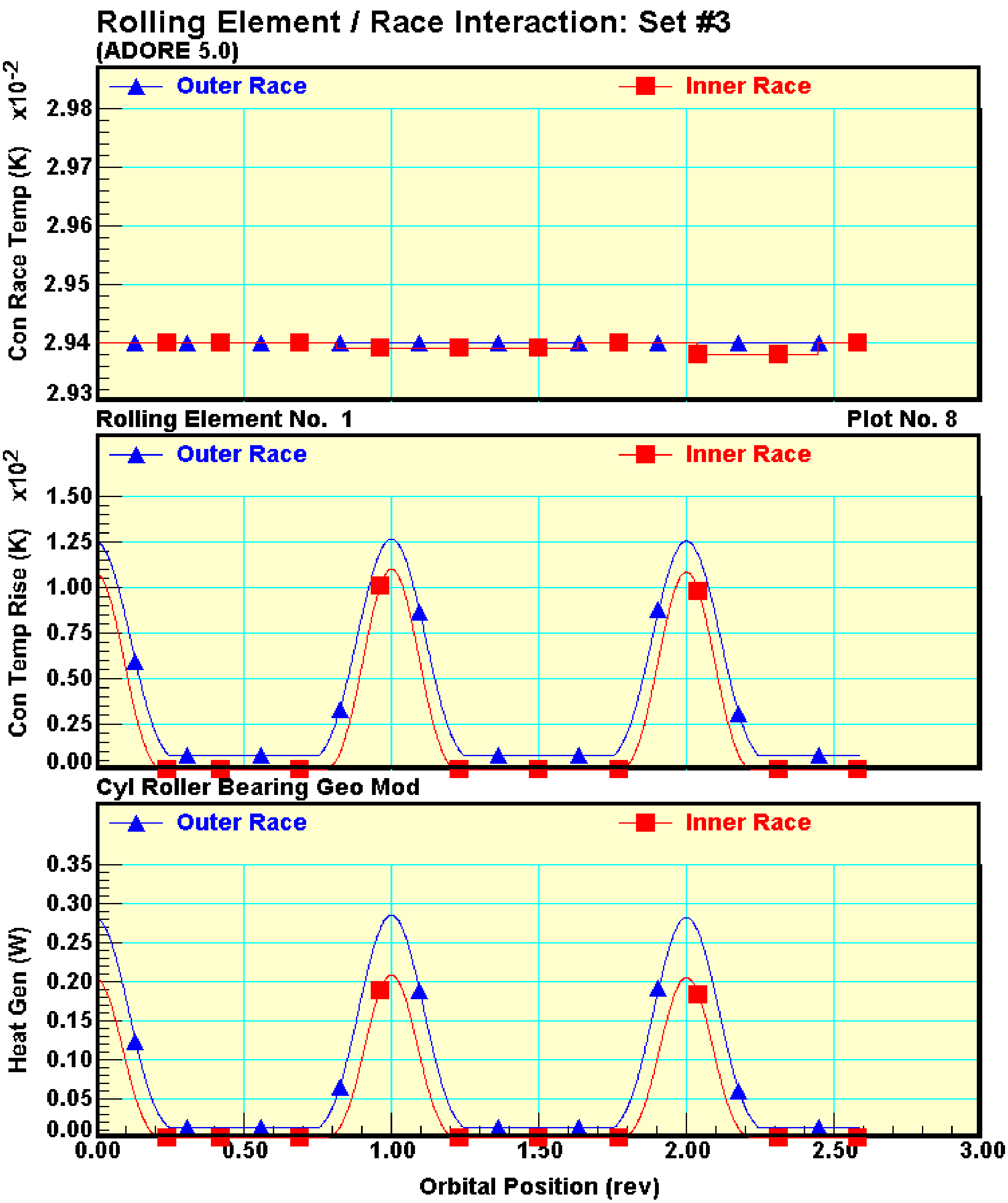

Figure 41. Typical roller/race contacts in the cylindrical roller bearing with changing bearing geometry due to thermal effects.

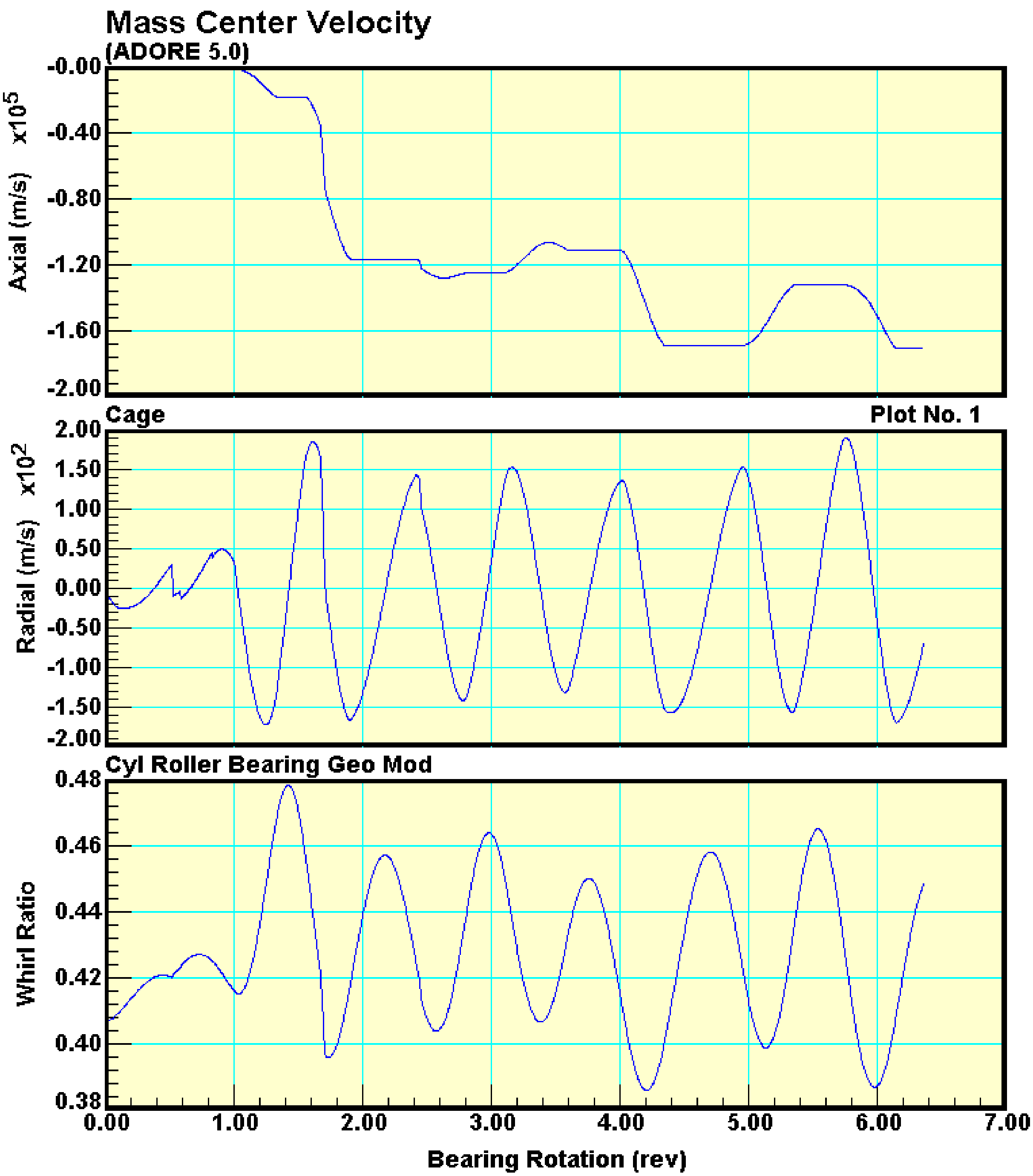

Figure 42. Change in cage mass center velocities in the roller bearing with changing bearing geometry as a result of thermal effects.

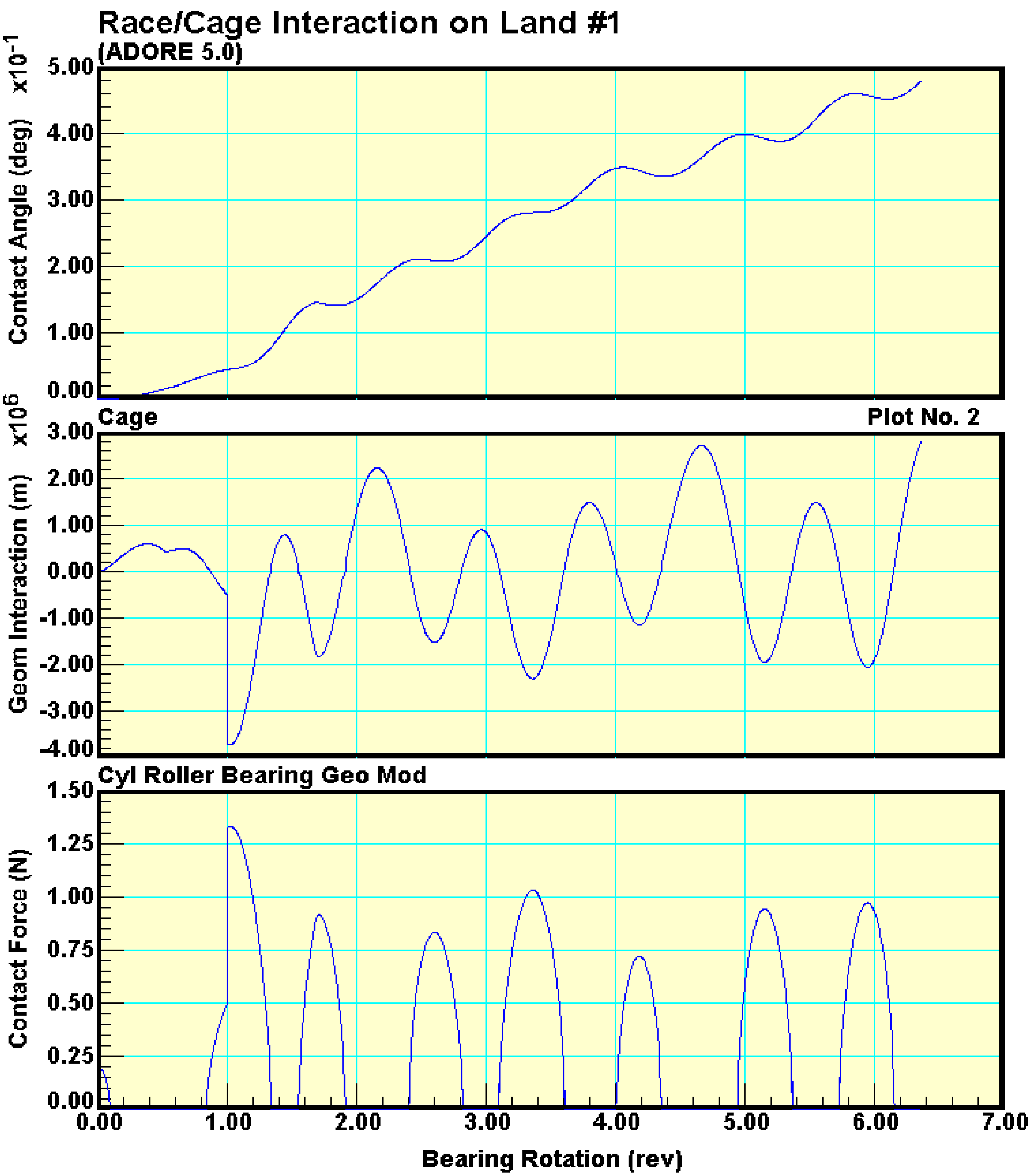

Figure 43. Increased cage/race forces in the roller bearing with thermal effects.

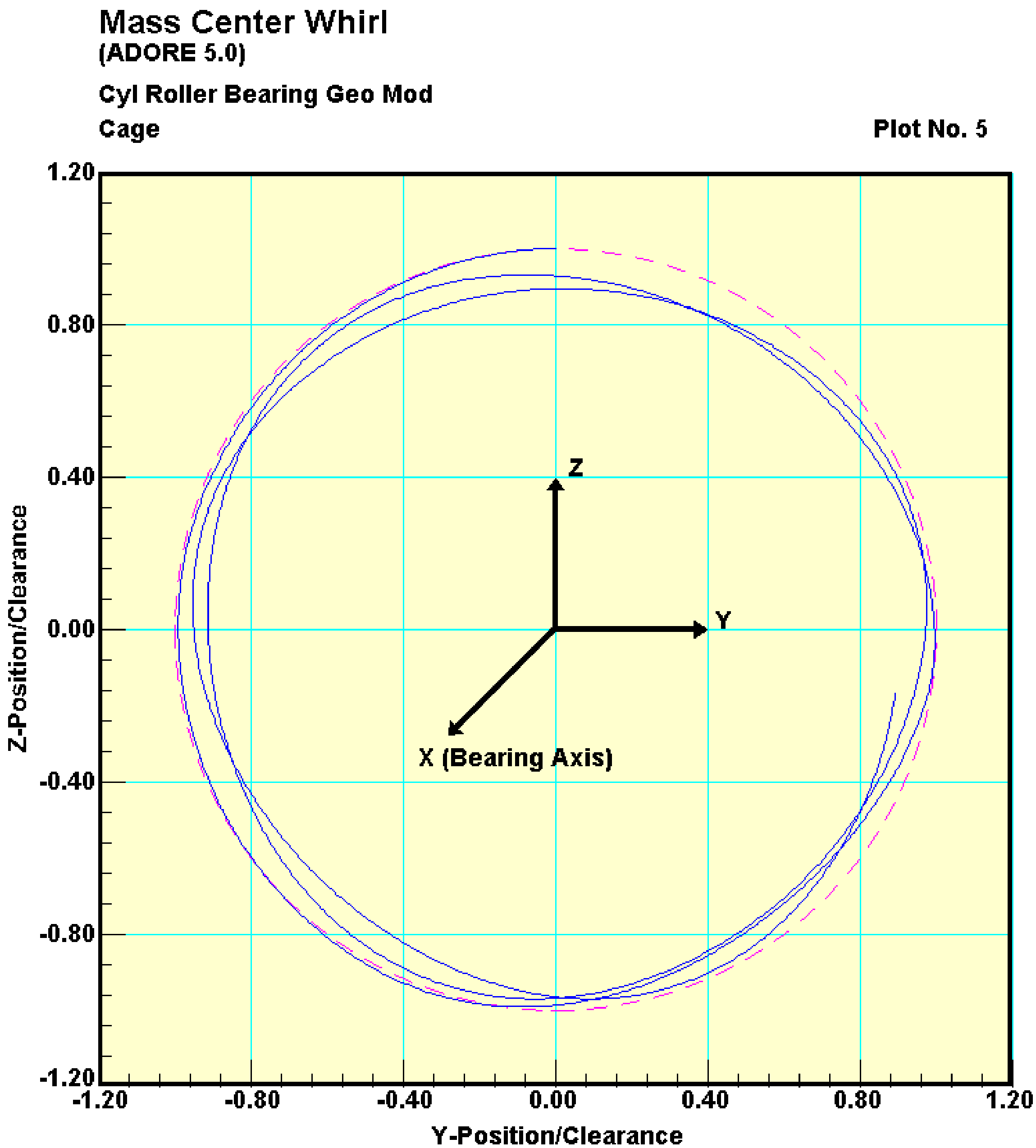

Figure 44. Cage mass center whirl in the roller bearing with thermal effects.

6. Conclusions

Based on the thermal modeling procedures developed in the current investigation and the preliminary parametric modeling of bearing performance it may be concluded that implementation of thermal interactions on overall bearing dynamics may be accomplished in three parts: first the computation of internal heat generation by precise and numerically accurate integration of mechanical interactions in the bearing; second computation of temperature field from transport of the internally generated heat to rest of the system; and finally an implementation of a change in bearing geometry and altered friction or traction conditions as a function of temperature. All of these steps, except changing traction with temperature, have been carried out in the present investigation. although implementation of time-varying traction models is straightforward, the required inputs require significant experimental development.

A step wise implementation of thermal effects seems to provide satisfactory results when the thermal model is free of any transients. The computed heat generation at each time step is averaged over a prescribed time interval before computing the thermal interactions and any changes in bearing geometry as a result of newly computed temperature field. In addition, any external constraint, such as equilibrium, may be imposed when the geometry is updated. Although such an implementation results in a step change in bearing performance parameters, no truncation errors in the integration of equations of motion are introduced when an explicit integration method is used. Thus the integrated solutions are numerically stable.

Parametric bearing performance simulations also show that when the traction characteristics are held constant the overall bearing performance as a function of changing internal geometry, due to thermal distortions, may not be significant if the bearing races are permitted to move and the applied loads are maintained at a constant value. Thus external systems constraints may be significant in thermal modeling of rolling bearing dynamics.

The computer models developed in the present investigation have been numerically tested for precision and convergent integration. The models are, therefore, very reliable for carrying out parametric or sensitivity studies to model overall effect of a given design parameter. However, significant experimental validation of the predicted results must be carried out before implementation of the computed results in practical designs.

7. Recommendations for Future Development

As a result of the modeling work carried out in the present investigation, several recommendations for further enhancements of the models may be made:

1. Experimental work leading to a more precise correlation of traction behavior with temperature is essential in overall rolling bearing dynamics. Once such data becomes available for potential lubricants, inputs to the current models can be significantly improved by varying traction as a function of computed temperatures.

2. Overall experimental validation of predicted results, such as bearing heat generation and computed temperatures fields is essential before the model can be reliably used for practical design and performance diagnosis. Since such investigations do not require measurement of any complicated motions within the bearings, they should be relatively easy to carry out.

3. Once thermally dependent traction is implemented and the model predictions have been experimentally validated, the model becomes an efficient tool for parametric studies leading classification of thermally induced phenomena, such as instabilities in bearing element motion. Such parametric studies may provide substantial guidance for practical bearing design and materials development.

4. The issue of thermal transients is another area where additional modeling work may be recommended. Since the time scales for thermal and mechanical interactions are greatly different, it may not be numerically efficient to integrate the generalized differential equations. However, once the models have some experimental validations, adequate number of parametric studies may be carried out to develop guidance for numerically averaging the thermal interactions, as arbitrarily carried out in the present investigation.

8. References

1. Kennel, J.W. and Puparia, S.S., “A Simplified Model of Cage Motion in Angular Contact Bearings Operating in the EHD Region”, J. Club. Tech., ASME Trans, Val 100, pp 395-403, 1978.

2. Gupta, P. K., “Some Dynamic Effects in a Solid-Lubricated Ball Bearing”, ASLE Trans., vol 26, pp 393-400, 1983.

3. Gupta, P. K., ADVANCED DYNAMICS OF ROLLING ELEMENTS, Springer-Verlag, 1984.

4. Gupta, P. K., “Frictional Instabilities in Ball Bearings”, STLE Tribology Trans, vol 31 (2), pp 258-268, 1988.

5. Gupta, P. K., “Frictional Instabilities in Roller Bearings”, STLE Tribology Trans, vol 33 (3), pp 395-401, 1990.

6. Gupta, P.K., “Some Dynamic Effects in High-S[peed Solid-Lubricated Ball Bearings”, ASLE Transactions, Vol 26, pp 393-400, 1983.

7. Abramowitz, M. and Stegun, I.A., HANDBOOK OF MATHEMATICAL FUNCTIONS, Dover Publications, 1968.

8. Tachman, E.G. and Cheng, H.S., NASA Technical Report CR-2206, March 1973.

9. Carlson, H. S. and Jaeger, J. C., CONDUCTION OF HEAT IN SOLIDS, Second Edition, Oxford Press, 1959.

10. MaAdams, W.H., HEAT TRANSMISSION, McGraw-Hill Book Company, 1954.

11. Kreith, F., PRINCIPLES OF HEAT TRANSFER, International Text Book Company, 1965.

12. Rohsenow, W.H., and Choi, H.Y., HEAT MASS AND MOMENTUM TRANSFER, Prentice Hall, 1961.

13. MARKS MECHANICAL ENGINEERS HANDBOOK, McGrawHill.

14. Gupta, P.K., “Thermal Interaction in Rolling Bearing Dynamics”, SBIR Phase I Technical Report G-119-TR-95, U. S. Air Force Contract F33615-95-C-2528, October 1995.

15. Den Hartog, J.P. ADVANCED STRENGTH OF MATERIALS, McGraw-Hill Book Company, 1952.

16. Tallian, T.E., “A Data-Fitted Rolling Bearing Life Prediction Model - Part I: Mathematical Model”, STLE Tribology Transactions, Vol 39, pp 249-258, 1996.

17. Tallian, T.E., “A Data-Fitted Rolling Bearing Life Prediction Model - Part II: Model Fit to the Historical Experimental Database”, STLE Tribology Transactions, Vol 39, pp 259-268, 1996.

18. Tallian, T.E., “A Data-Fitted Rolling Bearing Life Prediction Model - Part III: Parametric Study, Comparison to Published Models and Engineering Review”, STLE Tribology Transactions, Vol 39, pp 269-275, 1996.

19. Tallian, T.E., “A Data-Fitted Rolling Bearing Life Prediction Model - Part IV: Model Implementation for Current Engineering Use”, STLE Tribology Transactions, Vol 39, pp 957-963, 1996.

20. Tallian, T.E., “A Data-Fitted Rolling Bearing Life Prediction Model for Variable Operating Conditions”, STLE Tribology Transactions, Vol 42, pp 241-249, 1999.

21. Gupta, P.K., and Tallian, T.E., “Rolling Bearing Life Prediction Correction for Material and Operating Conditions - Part III: Implementation in Bearing Dynamics Computer Code”, ASME Journal of Lubrication Technology, Vol 112, pp 23-26, 1990.

22. Harris, T.A., ROLLING BEARING ANALYSIS, John-Wiley, 1984.

Appendix A - Typical ADORE Data Module

```
      module BrgGeom
!------------------------------------------------------------------------
!   This module contains all geometrical data for the bearing
!------------------------------------------------------------------------
      use Parameters
! overall bearing
!-----
      implicit none
      character*36                          :: runId              ! run identification string (max 36 chars)
      integer                               :: kBrg =0            ! bearing type, defined as
                                                                  ! 1 = ball bearing
                                                                  ! 2 = cylindrical roller bearing
                                                                  ! 3 = spherical roller bearing
                                                                  ! 4 = tapered roller bearing
                                                                  ! 5 = spherical tapered roller bearing
      integer,dimension(2,maxRe)            :: kConType =0        ! re/race contact type defined as follows --
                                                                  ! 0 = point contact (ball,sph roller/grooved race)
                                                                  !-1 = point contact (flat race/crowned roller)
                                                                  !-2 = point contact (crowned race/flat roller)
                                                                  !-3 = point contact (crowned race/crowned roller)
                                                                  ! 1 = line contact
      integer                               :: nRe =0             ! number of rolling elements
      real(r8)                              :: brgBore =0.        ! bearing bore (ID of inner race)
      real(r8)                              :: brgOD =0.          ! bearing outer diameter
      real(r8)                              :: shftID =0.         ! shaft inner diameter (hollow shaft)
      real(r8)                              :: hsngOD =0.         ! housing outer diameter
      real(r8)                              :: freeIntCls =0.     ! unmounted bearing internal clearance
      real(r8)                              :: freeConAng =0.     ! unmounted bearing contact angle (ball brgs only)
      real(r8)                              :: freeEndPlay        ! unmounted end play (ball bearings only)
      real(r8)                              :: pitchDia =0.       ! bearing pitch diameter
      integer                               :: nTE =0             ! total number of bearing elements (re+cage+race)
      real(r8)                              :: reAx               ! average of outer and inner race taper angles
      real(r8),dimension(3,3)               :: reBaseTran         ! initial transformation from inertial azimuth to
                                                                  ! rolling element coordinate frame, corresponding to
                                                                  ! initial orientation of rolling elements

!---------------------
! rolling element data
! ----
      real(r8)                              :: bReDia =0.         ! nominal rolling element diameter
                                                                  ! diameter at large end for tapered roller brgs
      real(r8),dimension(maxRe)             :: reDia =0.          ! rolling element diameter [1]
      real(r8),dimension(maxRe)             :: reDiaVar =0.       ! variation in rolling element diameter [1]
                                                                  ! variation in diameter at large end for tapered
                                                                  ! bearings
      real(r8),dimension(maxRe)             :: reRad =0.          ! rolling element radius [1] (mean rad for tapered brg)
      real(r8)                              :: bReLen =0.         ! nominal roller length
      real(r8),dimension(maxRe)             :: reLen =0.          ! roller length [1]
      real(r8),dimension(maxRe)             :: reLenVar =0.       ! variation in roller length [1]
      real(r8)                              :: bReCrn =0.         ! nominal roller crown radius
      real(r8),dimension(maxRe)             :: reCrn =0.          ! roller crown radius [1]
      real(r8),dimension(maxRe)             :: reCrnVar =0.       ! variation in roller crown [1]
      real(r8),dimension(2,maxRe)           :: reCurCen =0.       ! rolling element cur center (axial & radial comp)
                                                                  ! relative to rolling element geometric center
      real(r8)                              :: bReCenLen =0.      ! nominal central land length on roller
      real(r8),dimension(maxRe)             :: reCenLen =0.       ! central land half length on roller [1]
      real(r8),dimension(maxRe)             :: reCenLenVar =0.    ! variation in central land length on roller [1]
      real(r8),dimension(maxRe)             :: reCenLenOffset =0. ! axial offset of central land [1]
      real(r8),dimension(maxRe)             :: reCenCrnDrop =0.   ! crown drop on central land [1]
      real(r8),dimension(2,maxRe)           :: reConLmt           ! limit on major contact halfwidth [2]
      real(r8),dimension(2,maxRe)           :: reCenLmt           ! axial coordinate limits corresponding to central land
      real(r8),dimension(2,maxRe)           :: reCur =0.          ! rolling element curvatures [2]
      real(r8),dimension(2,maxRe)           :: rePrRad =0.        ! rad of deformed pressure surface at outer
                                                                  ! and inner race contact
      real(r8)                              :: bReTaper =0.       ! nominal semi taper angle of rollers
      real(r8),dimension(maxRe)             :: reTaper =0.        ! semi taper angle of rollers [1]
      real(r8),dimension(maxRe)             :: reTaperVar =0.     ! variation in semi taper angle of rollers [1]
      real(r8),dimension(maxRe)             :: reTanTaper =0.     ! tangent of roller taper angle [1]
      real(r8),dimension(maxRe)             :: reCosTaper =1._r8  ! cosine of roller taper angle [1]
      real(r8),dimension(2)                 :: bReCorRad =0.      ! nominal corner radii on neg & pos roller axis
      real(r8),dimension(2,maxRe)           :: reCorRad =0.       ! roller corner radii [3]
      real(r8),dimension(2,2,maxRe)         :: reCorCen =0.       ! roller corner center position [32]
      real(r8),dimension(2,maxRe)           :: reCorRadVar =0.    ! variation in roller corner radii [3]
      real(r8),dimension(2)                 :: bReEndRad =0.      ! nominal end radii on neg & pos roller axis
      real(r8),dimension(2,maxRe)           :: reEndRad =0.       ! roller end radii [3]
      real(r8),dimension(2,maxRe)           :: reEndCen =0.       ! axial position of roller end center [33]
      real(r8),dimension(2,maxRe)           :: reEndRadVar =0.    ! variation in roller end radii [3]
      real(r8),dimension(3,2,maxRe)         :: reEndFrame =0.     ! transformation angles for roller end face
                                                                  !  orientation [4]
      real(r8),dimension(3,3,2,maxRe)       :: tre                ! trans from roller geo to end face coords [34]
      real(r8)                              :: bReMass =0.        ! base mass (nominal rolling element mass)
      real(r8),dimension(3)                 :: bReMI =0.          ! base roll element moment of inertia
      real(r8),dimension(3)                 :: bReGeoCen =0.      ! base roll element geom center rel to mass center
```

```
      real(r8),dimension(3)                   :: bReFrame =0.        ! base roll element principal frame orientation
                                                                     ! principal frame rel to geometric frame
      real(r8),dimension(3,maxRe)             :: reGeoCen =0.        ! rolling element geom center rel to mass center[5]
      real(r8),dimension(maxRe)               :: reMass =0.          ! rolling element mass [1]
      real(r8),dimension(3,maxRe)             :: reMI =0.            ! rolling element moment of inertia [5]
      real(r8),dimension(3,maxRe)             :: reFrame =0.         ! orientation of rolling element principal axes [6]
      real(r8),dimension(3,3,maxRe)           :: reGeoToPrin         ! re geometrical to principal frame transformations
!-------------------
! race data
!-----
      real(r8),dimension(2)                   :: raceCurFac =0.      ! race curvature factors [7]
      real(r8),dimension(2)                   :: raceTaper =0.       ! race land semi taper angle [7]
      real(r8),dimension(2)                   :: raceTanTaper =0.    ! tangent of race semi taper angle [7]
      real(r8),dimension(2)                   :: raceCosTaper =1._r8 ! cosine of race semi taper angle (7)
      real(r8),dimension(2)                   :: raceCur =0.         ! race cross groove curvature (1/radius) [7]
      real(r8),dimension(2)                   :: raceCurRad =0.      ! race cross groove rad of cur for ball, spherial
                                                                     ! and tapered spherical roller bearings
      real(r8),dimension(2,2)                 :: raceCurCen =0.      ! axial and radial pos of race curvature center [12]
      real(r8),dimension(2)                   :: raceCrn             ! race surface crown used for cyl and tapered roller brgs
      real(r8),dimension(2)                   :: raceCenLen =0.      ! half length of central land on races [7]
      real(r8),dimension(2)                   :: raceCenCrnDrop =0.  ! race crown drop on central land [7]
      real(r8),dimension(2)                   :: raceLandLmt =0.     ! limiting length on race for roller interaction [7]
      real(r8),dimension(2)                   :: raceWidth =0.       ! width of races [7]
      real(r8),dimension(2,2)                 :: raceRad =0.         ! radius of race surface at mid point [31]
      real(r8),dimension(3,2)                 :: rlRadVar =0.        ! race out-of-round variation [35]
      real(r8),dimension(3,2)                 :: cFacVar =0.         ! race curvature factor variaion [35]
      real(r8),dimension(3,2)                 :: rlOffset =0.        ! race central land offset [35]
      real(r8),dimension(3,2)                 :: rlTaper =0.         ! race land taper variaton [35]
      real(r8),dimension(3,2)                 :: raceGeoCen =0.      ! race geometric center rel to it masscenter [9]
      real(r8),dimension(2)                   :: raceMass =0.        ! equivalent race mass [7]
      real(r8),dimension(3,2)                 :: raceMI =0.          ! race principal moment of inertia [9]
      real(r8),dimension(3,2)                 :: raceFrame =0.       ! orientation of race principal axes relative
                                                                     ! to its geometric frame [10]
      real(r8),dimension(3,3,2)               :: raceGeoToPrin       ! race geometrical to principal frame transformation
      integer,dimension(2)                    :: mRace(2)            ! race indices, element numbers
!-------------------
! guide flange data
!-----
      real(r8),dimension(2,2,2)               :: flngOrigin =0.      ! guide flange origin rel to race geo center [11]
      integer,dimension(2,2)                  :: kFlngInd =0         ! race flange identifier [13]
                                                                     ! (0=no flange, 1=flange present)
      integer,dimension(2)                    :: kFlngIndr =0        ! sum of flngInd for outer and inner races
      integer                                 :: kFlngInds =0        ! sum of all components of flngInd
      real(r8),dimension(2,2)                 :: flngHt =0.          ! guide flange height [13]
      real(r8),dimension(2,2)                 :: flngAng =0.         ! guide flange lay back angles [13]
      real(r8),dimension(2)                   :: flngCls =0.         ! flange clearance (cyl brg only) [7]
      real(r8),dimension(3,3,2,2)             :: traf                ! transformation from race azimuth to flange coordinates
!---------------------------
! cage data
!-----
      integer                                 :: nCseg =0            ! number of cage segments
      integer,dimension(maxCseg)              :: mCage               ! cage segment indices, element numbers
      integer                                 :: kPocType =0         ! cage pocket identifier
                                                                     ! for ball bearings:
                                                                     !  0=cylindrical,          1=spherical
                                                                     !  2=elongated cylindrical, 3=conical
                                                                     !  4=rectangular
                                                                     ! for roller bearings: 0=rectangular
                                                                     ! positive nonzero value, n, defines pairs of
                                                                     ! pocket interaction surfaces located symetrically
                                                                     ! relative to the pocket center, (maximum 5)
                                                                     ! =-1 cylindrical pocket, roller guided cage
      integer                                 :: kPocConType         ! pocket contact type, assigned in Adra2
      integer,dimension(maxRe,maxCseg)        :: nPocSur =1          ! number of guide surfaces in cage pocket
      real(r8),dimension(2,maxCseg)           :: cageDia =0.         ! cage outer and inner diameter
      real(r8),dimension(2,maxCseg)           :: cageRad =0.         ! outer and inner radius of cage
      real(r8),dimension(maxCseg)             :: cageWidth =0.       ! cage width
      real(r8)                                :: cageAngCut =0.      ! angular width of cut for segmented cage
      real(r8),dimension(maxCseg)             :: cageConeAng =0.     ! cage semi-cone angle for tapered roller bearings
      integer                                 :: nGL                 ! number of cage/race guide lands, maximum value set
                                                                     ! by parameter maxGL in the Parameters module
      integer, dimension(maxGL,maxCseg)       :: iCageGuide =0       ! cage guidance type (1=outer, 2=inner) [14]
      real(r8),dimension(maxGL,maxCseg)       :: cageGsDia =0.       ! diameters at cage guide surface [14]
      real(r8),dimension(maxGL,maxCseg)       :: cageGsRad =0.       ! radius at cage guide surface [14]
      real(r8),dimension(maxGL,maxCseg)       :: cageGsCls =0.       ! cage guide surface diametral clearances [14]
      real(r8),dimension(maxGL,maxCseg)       :: cageGsWidth =0.     ! cage guide surface width [14]
      real(r8),dimension(maxGL,maxCseg)       :: cageGsPos =0.       ! axial pos of outer edge of guide surface from cage center
      real(r8),dimension(maxCseg)             :: cageGsClss =0.      ! inverse of cls value used to scale mass center position
      real(r8),dimension(2,maxCseg)           :: bPocCls =0.         ! nominal diametral pocket clearances
      real(r8),dimension(maxGL,maxCseg)       :: raceGsRad =0.       ! race guide surface radius
      real(r8),dimension(3,maxGL,maxCseg)     :: cageGsRadVar =0.    ! cage guide land radius variation (amp,freq & phase)
      real(r8),dimension(3,maxGL,maxCseg)     :: raceGsRadVar =0.    ! race guide land radius variation (amp,freq & phase)
      real(r8),dimension(2,maxRe,maxCseg)     :: pocCls =0.          ! radial (diametral/2) pocket clearances [15]
      real(r8),dimension(2,maxRe,maxCseg)     :: pocClsVar =0.       ! variation in pocket diametral clearances [15]
      real(r8),dimension(2,maxCseg)           :: cageCls =0.         ! cage outer and inner diametral clearances for churning
                                                                     !  calculations
      real(r8),dimension(maxRe,maxCseg)       :: pocRad =0.          ! cage pocket radius [15]
                                                                     ! or circumferential pocket width for rect pockets
      real(r8),dimension(maxRe,maxCseg)       :: pocWidth =0.        ! axial pocket width for rect pockets [15]
      real(r8),dimension(3,maxPgs,maxCseg)    :: bPocGsAng =0.       ! nominal pocket guide surface orientation relative to
                                                                     ! nominal pocket surface orientation [16]
      real(r8),dimension(3,maxPgs,maxCseg)    :: bPocGsCen =0.       ! nominal coordinates of pocket guide surface center
```

```
                                                                     ! rel to nominal pocket center [17]
   real(r8),dimension(2,maxPgs,maxCseg)       :: bPocGsLen =0.       ! nominal pocket guide surface lengths[18]
   real(r8),dimension(3,maxPgs,maxRe,maxCseg) :: pocGsAng =0.        ! pocket guide surface angles relative to nominal
                                                                     ! pocket surface orientation [19]
   real(r8),dimension(3,3,maxPgs,maxRe,maxCseg) :: pocToGs = 0.      ! cage pocket to guide surface transformation
   real(r8),dimension(3,maxPgs,maxRe,maxCseg) :: pocGsAngVar=0.      ! variations in pocket guide surface angles [19]
   real(r8),dimension(3,maxPgs,maxRe,maxCseg) :: pocGsCen =0.        ! coordinates of guide surface center [20]
   real(r8),dimension(3,maxPgs,maxRe,maxCseg) :: pocGsCenVar=0.      ! variation in coordinates of guide surface
                                                                     ! center [20]
   real(r8),dimension(2,maxPgs,maxRe,maxCseg) :: pocGsLen =0.        ! width of pocket guide surfaces [21]
   real(r8),dimension(2,maxPgs,maxRe,maxCseg) :: pocGsLenVar=0.      ! variation in with of pocket guide surfaces [21]
   real(r8),dimension(2,4,maxPgs,maxRe,maxCseg) :: pocGsCor =0.      ! pocket guide surface corner coordinates (x & z comps)
   real(r8),dimension(3,maxRe,maxCseg)        :: pocCen =0.          ! pocket center relative to cage geometric
                                                                     ! center [22]
   real(r8),dimension(3,maxRe,maxCseg)        :: pocAng =0.          ! pocket orientation [23]
   real(r8),dimension(maxRe,maxCseg)          :: pocConeAng          ! semi cage pocket cone angle for conical pockets
   real(r8),dimension(maxRe,maxCseg)          :: sPocConeAng         ! sine of semi cage pocket cone angle
   real(r8),dimension(maxRe,maxCseg)          :: cPocConeAng         ! cosine of semi cage pocket cone angle
   real(r8),dimension(3,3,maxRe,maxCseg)      :: cageToPoc           ! cage to pocket transformation
   real(r8),dimension(maxRe,maxCseg)          :: pocThkns =0.        ! pocket thickness along the radial direction [24]
   real(r8),dimension(maxRe,maxCseg)          :: pocThknsVar =0.     ! variation in pocket thickness
   real(r8),dimension(maxRe,maxCseg)          :: pocWidthVar =0.     ! variation in pocket width in axial direction
   real(r8),dimension(2,maxRe,maxCseg)        :: pocCenVar =0.       ! variation in axial & circum coordinate of pocket center
   real(r8),dimension(3,maxRe,maxCseg)        :: pocAngVar =0.       ! variation in pocket angular orientation (y & z rots)
   real(r8),dimension(2,maxRe,maxCseg)        :: pocLimT =0.         ! angles theta defining pocket surface limits [25]
   real(r8),dimension(2,maxRe,maxCseg)        :: pocLimP =0.         ! angles phi defining pocket surface limits
   real(r8),dimension(2,maxRe,maxCseg)        :: pocLimZ =0.         ! z-coordinate limits defining pocket surface limits
   real(r8),dimension(maxRe,maxCseg)          :: pocConLimB =0.      ! limit on contact half width for roller brg pockets
   real(r8)                                   :: bCageMass =0.       ! base cage segment mass
   real(r8),dimension(maxCseg)                :: cageMass =0.        ! mass of cage segment [26]
   real(r8),dimension(3,maxCseg)              :: cageGeoCen =0.      ! cage geometric center relative to its mass
                                                                     ! center [27]
   real(r8),dimension(3,maxCseg)              :: cageFrame =0.       ! orientation cage princial frame rel to its
                                                                     ! geometric frame [28]
   real(r8),dimension(3,3,maxCseg)            :: cageGeoToPrin       ! cage geometric to principal frame transformation
   real(r8),dimension(3)                      :: bCageMI =0.         ! base cage moment of inertia
   real(r8),dimension(3,maxCseg)              :: cageMI =0.          ! cage principal moment of intertia [29]
   integer, dimension(maxRe,maxCseg)          :: iPocRe =0           ! rolling element identifier for a pocket [30]
   integer, dimension(maxCseg)                :: nPoc =0             ! number of pockets in cage segments
   real(r8),dimension(maxCseg)                :: csegLen =0.         ! cage segment length in degrees
   real(r8),dimension(maxRe,maxCseg)          :: pocFrac             ! extent of effective fractional pocket area for
                                                                     ! for rolling element interaction  1.0=full pocket

!----------------------------------------
! NOTES
! -----
!
!  Description of dimensional indices in the above arrays is indicated by a note number is squares brackets. These
!  notes are documented in the table below:
!
!  Note #    Dim Index #    Max Limit    Index Description
!  ------    -----------    ---------    -----------------
!     1          1           nRe         rolling element #
!     2          1           2             along and normal to rolling direction
!                2           nRe         rolling element #
!     3          1           2             negative and positive roller axis
!                2           nRe         rolling element #
!     4          1           3             x,y,z rotations
!                2           2             negative and positive roller axis
!                3           nRe         rolling element #
!     5          1           3             x,y,z components
!                2           nRe         rolling element #
!     6          1           3             x,y,z rotations
!                2           nRe         rolling element #
!     7          1           2             race # (1=outer, 2=inner)
!     7a         1           2             axial and radial position
!                2           2             race # (1=outer, 2=inner)
!     8          1           2             along and normal to rolling direction
!                2           2             race # (1=outer, 2=inner)
!     9          1           3             x,y,z components
!                2           2             race # (1=outer, 2=inner)
!    10          1           3             x,y,z rotations
!                2           2             race # (1=outer, 2=inner)
!    11          1           2             axial and radial components
!                2           2             negative and positive x-axis
!                3           2             race # (1=outer, 2=inner)
!    12          1           2             axial and radial components
!                2           2             race # (1=outer, 2=inner)
!    13          1           2             negative and positive x-axis
!                2           2             race # (1=outer, 2=inner)
!    14          1           2             negative and positive x-axis
!    15          1           nRe         pocket #
!                2           nCseg       cage segment #
!    16          1           3             x,y,z rotations
!                2           numPgs        pocket guide surface #
!    17          1           3             x,y,z components
!                2           numPgs        pocket guide surface #
!    18          1           2             normal and along roller axis
!                2           numPgs        guide surface #
!    19          1           3             x,y,z rotations
!                2           numPgs        guide surface #
!                3           nRe         pocket #
```

```
!              4            nCseg       cage segment #
!   20         1            3             x,y,z components
!              2            numPgs        guide surface #
!              3            nRe         pocket #
!              4            nCseg       cage segment #
!   21         1            2             normal and along roller axis
!              2            numPgs        guide surface #
!              3            nRe         pocket #
!              4            nCseg       cage segment #
!   22         1            3             x,y,z components
!              2            nRe         pocket #
!              3            nCseg       cage segment #
!   23         1            3             x,y,z rotations
!              2            nRe         pocket #
!              3            nCseg       cage segment #
!   24         1            nRe         pocket #
!              2            nCseg       cage segment #
!   25         1            2             two angles, rotaton about z and x-axes
!              2            nRe         pocket #
!              3            nCseg       cage segment #
!   26         1            nCseg       cage segment #
!   27         1            3             x,y,z components
!              2            nCseg       cage segment #
!   28         1            3             x,y,z rotations
!              2            nCseg       cage segment #
!   29         1            3             x,y,z components
!              2            nCseg       cage segment #
!   30         1            nRe         pocket #
!              2            nCseg      cage segment #
!   31         1            2            race radius at mid point of contact surface (1=outer, 2=inner)
!              2            2            race # (1=outer, 2=inner)
!   32         1            2          axial position of roller corner center rel to geo center (1=neg axis, 2=pos axis)
!              2            2          radius of roller corner center locus (1=neg axis, 2=pos axis)
!              3            maxRe      rolling element index
!   33         1            2          axial position of roller end center rel to geo center (1=neg axis, 2=pos axis)
!              2            maxRe      rolling element index
!   34         1,2          3          transformation matrix from roller geo to end face coordinate system
!              3            2          1=neg axis, 2=pos axis
!              4            maxRe      rolling element index
!   35         1            3          1=magnitude, 2=frequency of variation and 3=phase shift from principal frame
!              2            2          1=outer race, 2=inner race.
!--------------------------------------------------
      end module BrgGeom
```

Appendix B - Typical ADORE Code Segment

```
      subroutine Adrc1
!
!-----------------------------------------------------------------------
!    Adrc1   rolling element/race normal load
!-----------------------------------------------------------------------
!
      use Parameters
      use BrgGeom
      use Constants
      use VecMats
      use ReVecs
      use LoadDer
      use Solutions
      use OpCond
      use SubX
      use TmpArea                                               ! used in LineContact
      use FatigueCons                                           ! used in LineContact
      use Errors                                                ! used in NewContLmts
      use ErrorCodes                                            ! used in NewContLmts
      use FlngData                                              ! used in FlngContact
      use SetUpB1                                               ! store results for use in AdrB1
      implicit none
!-----------------------------------
!     local variables:
      integer                             :: j                  ! race index
      integer                             :: k,l,ll             ! loop indices
      real(r8),dimension(3,3)             :: tib                ! inertial to re frame transformation
      real(r8),dimension(3,3)             :: tbra               ! roll ele to race azimuth transformation
      real(r8),dimension(3)               :: bgi                ! re geo center in intertial frame
      real(r8),dimension(3)               :: rgi                ! race geo center in intertial frame
      real(r8),dimension(3)               :: bgrgi              ! re geo center rel to race geo center in intertial frame
      real(r8),dimension(3)               :: bgrgra             ! re geo center rel to race geo center in race azimuth frame
      real(r8),dimension(3)               :: rcrgra             ! race cur center rel to race geo center in race azimuth frame
      real(r8),dimension(3)               :: bcbgr              ! re cur center rel to re geo center in race frame
      real(r8),dimension(3)               :: bcbgb              ! re cur center rel to re geo center in re frame
      real(r8),dimension(3)               :: bcbgra             ! re cur center rel to re geo center in race azimuth frame
      real(r8),dimension(3)               :: bcrcra             ! re cur center rel race cur center in race azimuth frame
      real(r8)                            :: sbsi,cbsi          ! sine and cosine of inertial azimuth angle
      real(r8)                            :: si                 ! azimuth angle
      real(r8)                            :: ssi,csi            ! sine and cosine of azimuth angle
      real(r8)                            :: rv                 ! race radius variation
      real(r8)                            :: rcfv               ! race curvature factor variation
      real(r8)                            :: a1,a2              ! local variables
      real(r8)                            :: Adrb3,Adrx6        ! define function types
!-------------------------------------------------
!     these variables are computed in LCDerivatives
!     and used in LineContact procedure
!     --------------------------------
      real(r8),dimension(4)               :: delQbj             ! derivative of load with roller displacements
                                                                ! x, r, ang2 and ang3
      real(r8),dimension(4)               :: delMbj             ! derivaatie of moment with roller displacements
                                                                ! x, r, ang2 and ang3
      real(r8),dimension(5)               :: delQrj             ! derivative of load with race displacements
                                                                ! X, Y, Z, ang2 and ang3
      real(r8),dimension(5)               :: delMrj             ! derivative of moment with race displacements
                                                                ! X, Y, Z, ang2 and ang3
!-------------------------------------
!
!     rolling element geo center and base transformations
!     ------------------------------------------------
!
       tia=inertToIaz(:,:,ire)                                  ! inertToIaz,inertToRe,reToIaz->VecMats
       cbsi=tia(3,3)
       sbsi=tia(2,3)
       bgi=reGci(:,ire)                                         ! roll ele geo center in intertial frame
       tib=inertToRe(:,:,ire)                                   ! inertial to rolling element frame transformation
       tbia=reToIaz(:,:,ire)                                    ! roll ele to azimuth frame transformationation, tbia->ReVecs
       if (ider /= 0) then                                      ! ider->LoadDer
          call Adrb22(sinBt(:,ire),cosBt(:,ire),tib2)           ! sinBt,cosBt->ReVecs
          call Adrb23(sinBt(:,ire),cosBt(:,ire),tib3)
       end if
!
!     start race loop
!     --------------
!
      idFlngSol=0                                               ! initialize flange solution indicator, idFlngSol->Solutions
      reEndWR(1,ire)=zero                                       ! initialize roller end wear, reEndWR->Solutions
      reEndWR(2,ire)=zero
      reFlngFor=zero                                            ! flange force vector on roll elements, reFlngFor->ReVecs
      reFlngMom=zero                                            ! flange moment vector on roll elements, reFlngMom->ReVecs
      raceFlngFor=zero                                          ! flange force vector on races, raceFlngFor->ReVecs
      raceFlngMom=zero                                          ! flange moment vector on races, raceFlngMom->ReVecs
```

```
      bFlngFor=zero                                                     ! normal flange force vector on roller, bFlngFor->ReVecs
      bFlngMom=zero                                                     ! normal flange moment vector on roller, bFlngMom->ReVecs
      rFlngFor=zero                                                     ! normal race flange force vector, rFlngFor->ReVecs
      rFlngMom=zero                                                     ! normal race flange moment vector, rFlngMom->ReVecs
      dbFlngFor=zero                                                    ! roller flange force derivatives, dbFlngFor->LoadDer
      dbFlngMom=zero                                                    ! roller flange mment derivatives, dbFlngMom->LoadDer
      drFlngFor=zero                                                    ! race flange force derivatives, drFlngFor->LoadDer
      drFlngMom=zero                                                    ! race flange moment derivatives, drFlngMom->LoadDer
      do j=1,2
         irace=j                                                        ! irace->ReVecs
!
!        race centers and transformations
!        --------------------------------
!
         rgi(:)=raceGc(:,irace)                                         ! raceGc->VecMats
         tir(:,:)=inertToRace(:,:,irace)                                ! inertToRace->VecMats, tir->ReVecs
         tbr=matmul(tir,transpose(tib))                                 ! roll element to race coordinate frame transformation
                                                                        ! tbr->ReVecs
         tria=raceToIaz(:,:,j,ire)                                      ! race to inertial azimuth frame transformation
                                                                        ! raceToIaz->VecMats
         if (ider /= 0) then                                            ! ider->LoadDer
            tir2=inertToRace2(:,:,j)                                    ! inertToRace2,inertToRace3->LoadDer
            tir3=inertToRace3(:,:,j)
         end if
!
!        rolling element relative to race
!        --------------------------------
!
         bgrgi=bgi-rgi                                                  ! roll element geo center relative to race geo center
         bgrgr=matmul(tir,bgrgi)                                        ! in race frame, bgrgr->ReVecs
         tra=zero                                                       ! zero->Constants, tra->ReVecs
         si=Adrb3(bgrgr(2),bgrgr(3))                                    ! race azimuth
         sii=si                                                         ! sii->ReVecs, store azimuth for use in Adrc3
         ssi=dsin(si)
         csi=dcos(si)
         tra(1,1)=one
         tra(2,2)=csi
         tra(3,3)=csi
         tra(2,3)=ssi
         tra(3,2)=-ssi
         tras(:,:,j)=tra                                                ! tras,tra->ReVecs
         do k=1,3
            bgrgra(k)=zero
            do l=1,3
               bgrgra(k)=bgrgra(k)+tra(k,l)*bgrgr(l)                    ! re geo cen rel to race geo cen in race azimuth frame
            end do
         end do
!
!        base point on race for roll ele interaction
!        -------------------------------------------
!
         if (jcm(6) == 0) then                                          ! jcm->SubX
            rv=zero                                                     ! zero->Constants, race radius variation
            rcfv=zero                                                   ! groove curvature factor variation
         else
            rv=Adrx6(irace,rcfv,si)
         end if
         rcrgra(1)=raceCurCen(1,j)                                      ! raceCurCen->BrgGeom
         rcrgra(2)=zero                                                 ! for flat race, raceCurCen is zero
         rcrgra(3)=raceCurCen(2,j)
         if (raceCurRad(j) > crnLmt) then                               ! set base point to race land mid point
            if (irace == 1) then                                        ! for flat race surface
               rcrgra(3)=raceRad(2,irace)+rv                            ! race rad is base for line contact
            else                                                        ! raceRad->BrgGeom
               rcrgra(3)=raceRad(1,irace)+rv
            end if
         else if (raceCenLen(j) > zero) then                            ! set base point to race land mid point
            if (irace == 1) then                                        ! for partly crowned race surface
               rcrgra(3)=raceRad(2,irace)+rv                            ! race rad is base for line contact
            else                                                        ! raceRad->BrgGeom
               rcrgra(3)=raceRad(1,irace)+rv
            end if
         end if
!
!        re cur center relative to re geo center
!        ---------------------------------------
!
         if (kConType(j,ire) > 0) then
            bcbgra=zero                                                 ! for line contact this vector is zero
         else
            tbra=matmul(tra,tbr)                                        ! re to race azimuth transformation
            si=datan(tbra(2,3)/tbra(2,2))                               ! compute angular position of contact on re
            if (reCurCen(1,ire) /= zero) then
               a1=tbra(2,1)*reCurCen(1,ire)/(reCurCen(2,ire)            &
     &            *dsqrt(tbra(2,2)**2+tbra(2,3)**2))
               a2=dsqrt(one-a1*a1)
               si=si+datan(a1/a2)
            end if
            bcbgb(1)=reCurCen(1,ire)
            bcbgb(2)=-reCurCen(2,ire)*dsin(si)
            bcbgb(3)=reCurCen(2,ire)*dcos(si)
            a1=tbra(3,1)*bcbgb(1)+tbra(3,2)*bcbgb(2)+tbra(3,3)*bcbgb(3) ! z oomponent of cur cen location in azimuth frame
```

```
            if ((sig(j)*a1) > zero) then
               bcbgb(3)=-bcbgb(3)
               bcbgb(2)=-bcbgb(2)
            end if
            bcbgra=matmul(tbra,bcbgb)                                    ! re cur center in race azimuth frame
         end if
         bcbgr=matmul(transpose(tra),bcbgra)                             ! this vector is used later to define contact position
!
!        re cur center rel to race base point
!        ------------------------------------
!
         bcrcra=bgrgra+bcbgra-rcrgra
!
!        call appropriate load module
!        ----------------------------
!
         if (kConType(irace,ire) <= 0) then                              ! kConType->BrgGeom
            call PointContact                                            ! point contact solution
         else
            call LineContact                                             ! line contact solution
         end if
         if (kFlngIndr(j) /= 0) then                                     ! kFlngIndr->BrgGeom
            call FlngContact                                             ! flange contact for roller bearings
            raceFlngFors(:,j,ire)=raceFlngFor(:,j)                       ! stored for later use in Adrb1
            raceFlngMoms(:,j,ire)=raceFlngMom(:,j)                       ! stored for later use in Adrb1
         end if                                                          ! raceFlngFors,raceFlngMom2->SetUpB1
      end do
      if (kFlngInds /= 0) then                                           ! kFlngInds->BrgGeom
!-----------------------------------------------
!        store re flange results for future use
!        --------------------------------------

         reFlngFors(:,ire)=reFlngFor                                     ! reFlngFors->SetUpB1, stored for Adrb1
         reFlngMoms(:,ire)=reFlngMom                                     ! reFlngMoms->SetUpB1, stored for Adrb1
      end if
      return
      contains
         subroutine PointContact
!
!-----------------------------------------------------------------
!        point contact solutions
!-----------------------------------------------------------------
!
         integer                             :: i,j                      ! roll ele # and race #
         integer                             :: k,l                      ! loop indices
         real(r8),dimension(3)               :: uvcon                    ! unit vector along load in contact frame
         real(r8),dimension(3)               :: uvra                     ! unit load vector in race azimuth frame
         real(r8),dimension(3)               :: uvr                      ! unit load vector in race frame
         real(r8),dimension(3)               :: uvia                     ! unit load vector in intertial azimuth frame
         real(r8)                            :: tol =1.0E-05_r8          ! tolerance for out-of-plane contact
         real(r8)                            :: pRad                     ! race radius at contact point
         real(r8)                            :: cal                      ! cosine of contact angle
         real(r8)                            :: tal                      ! tangent of contact angle
         real(r8)                            :: ang                      ! zero or pi, used in contact angle computation
         real(r8)                            :: sa1,sa2,ca1,ca2          ! sine and cosine of computed contact angles
         real(r8)                            :: bcrca                    ! magnitude of vector bcrcra
         real(r8)                            :: rrx,rry                  ! race curvatures along and normal to rolling dir
         real(r8)                            :: sum, fr                  ! curvature sum and function for load computation
         real(r8)                            :: aj,bj,dj,aa              ! local variables used for contact load computation
         real(r8)                            :: efl,bl,dq,ql             ! local variables for line contact computation
         real(r8),dimension(2)               :: pqs                      ! elastic constants for line contact computation
         real(r8),dimension(3)               :: bcrcc                    ! re cur center rel race center in contact frame


!
!        initialize solutions
!        --------------------
!
         i=ire                                                           ! ire->ReVecs
         j=irace                                                         ! irace->ReVecs
         dqt(j)=zero                                                     ! dqt->LoadDer, zero->Constants
         dmt(j)=zero                                                     ! dmt->LoadDer
         dmx(j)=zero                                                     ! dmx->LoadDer
         dmr(j)=zero                                                     ! dmr->LoadDer
         dqx(j)=zero                                                     ! dqx->LoadDer
         dqr(j)=zero                                                     ! dqr->LoadDer
         dqxx(j)=zero                                                    ! dqxx->LoadDer
         dqyy(j)=zero                                                    ! dqyy->LoadDer
         dqzz(j)=zero                                                    ! dqzz->LoadDer
!
!        geo interaction and unit load vector
!        ------------------------------------
!
         uvcon(1)=zero                                                   ! unit vector along load in contact frame
         uvcon(2)=zero                                                   ! zero->Constants
         uvcon(3)=one
         pRad=rcrgra(3)                                                  ! base radius for rec/race contact
         if(dabs(raceCurRad(j)) > crnLmt) then                           ! raceCurRad->BrgGeom, crnLmt->Constants
!
!           flat race
!           ---------
!
```

```
          tac(:,:)=tacs(:,:,j,i)                                      ! race azimuth to contact frame transformation
                                                                      ! tac->ReVecs, tacs->VecMats
          bcrcc=matmul(tac,bcrcra)                                    ! re cur cen rel to race land in contact frame
          conDef(j,i)=bcrcc(3)+reCrn(i)                               ! geometric interaction, del->Solutions, reCrn->BrgGeom
          uvra=matmul(transpose(tac),uvcon)                           ! unit vector along load in race azimuth frame
          uvr=matmul(transpose(tra),uvra)                             ! in race frame, tra->ReVecs
          uvia=matmul(tria,uvr)                                       ! in inertial azimuth frame
          cal=tac(1,1)                                                ! tac->ReVecs
       else if(dabs(reCrn(i)) > crnLmt) then                          ! reCrn->BrgGeom, crnLmt->Constants
!
!         flat re surface
!         ---------------
!
!         to be formulated later
          continue
       else
!
!         both surfaces are curved
!         ------------------------
!
          bcrca=dsqrt(bcrcra(1)**2+bcrcra(2)**2+bcrcra(3)**2)
          conDef(j,i)=bcrca+reCrn(i)-raceCurRad(j)                    ! reCrn->BrgGeom, raceCurRad->BrgGeom
          uvra(1)=bcrcra(1)/bcrca
          uvra(2)=bcrcra(2)/bcrca
          uvra(3)=bcrcra(3)/bcrca
          uvr=matmul(transpose(tra),uvra)                             ! in race frame, tra->ReVecs
          uvia=matmul(tria,uvr)                                       ! in inertial azimuth frame
          cal=uvra(3)
!
!         contact angles
!         --------------
!
          ang=(j-1)*pi                                                ! pi->Constants
          tal=uvia(1)/uvia(3)                                         ! tangent of contact angle
          if(dabs(tal).le.tolr(2)) then                               ! tolr->Constants
             conAng(j,1,i)=ang                                        ! conAng->Solutions
          else
             conAng(j,1,i)=ang+datan(tal)                             ! conAng->Solutions
          end if
          if(dabs(uvia(2)) <= tol) then
             conAng(j,2,i)=zero                                       ! conAng->Solutions, zero->constants
          else
             conAng(j,2,i)=datan(-uvia(2)                             & ! conAng->Solutions
     &          /dsqrt(uvia(1)**2+uvia(3)**2))
          end if
          sa1=dsin(conAng(j,1,i))                                     ! inertial azimuth to contact frame transformation
          sa2=dsin(conAng(j,2,i))                                     ! tac->ReVecs
          ca1=dcos(conAng(j,1,i))
          ca2=dcos(conAng(j,2,i))
          tac(1,1)=ca1
          tac(2,1)=sa1*sa2
          tac(3,1)=sa1*ca2
          tac(1,2)=zero
          tac(2,2)=ca2
          tac(3,2)=-sa2
          tac(1,3)=-sa1
          tac(2,3)=ca1*sa2
          tac(3,3)=ca1*ca2
       end if
       tacs(:,:,j,i)=tac
!
!      contact load and half widths
!      ----------------------------
!
       if(conDef(j,i) > tolr(1)) then                                 ! tolr->Constants, conDef->Solutions
          if(dabs(raceCurRad(j)) < crnLmt) then                       ! raceCurRad->BrgGeom, crnLmt->Constants
             rrx=-one/(pRad/cal+raceCurRad(j))
             rry=-one/raceCurRad(j)
          else
             rrx=-cal/pRad
             rry=zero
          end if

          radX(j)=one/(rrx+reCur(1,i))                                ! eq rad along x direc (rolling direc), radX->ReVecs
                                                                      ! radX is used in traction procs
          sum=rrx+rry+reCur(1,i)+reCur(2,i)                           ! reCur->BrgGeom, i->ReVecs
          fr=(rrx+reCur(1,i)-rry-reCur(2,i))/sum
          if(fr <= 0.9995_r8) then
             call Adrc4(fr,aj,bj,dj,0)
             if (ierInd /= 0) return
             conLoad(j,i)=(two*sum/three)*(two*conDef(j,i)            & ! conLoad,del->Solutions, two,three,one2->constants
     &          /(dj*sum))**(one2*three)
             aa=(one2*three*conLoad(j,i)/sum)**one3                   ! one3->Constants
             conWidthA(j,i)=aj*aa                                     ! conWidthA->Solutions
             conWidthB(j,i)=bj*aa                                     ! conWidthB->Solutions
             conLoad(j,i)=conLoad(j,i)/qmReRace(j)                    ! conLoad->Solutions, qmReRace->Constants
             conPress(j,i)=three*conLoad(j,i)/(two*pi*conWidthA(j,i)  & ! conPress->Solutions
     &                     *conWidthB(j,i))
             dqn(j)=one2*three*conLoad(j,i)/conDef(j,i)               ! dqn->LoadDer
          else
             efl=two*reConLmt(j,i)                                    ! line contact with predefined length when
             conWidthA(j,i)=one2*efl                                  ! curvature function is close to 1.0, bl->BrgGeom
```

```
           pqs(1)=qReRace(1,j)                                         ! conWidthA->Solutions
           pqs(2)=qReRace(2,j)                                         ! qReRace->Constants
           call Adrc6(1,conDef(j,i),pqs,efl,sum,bl,dq,ql)
           conLoad(j,i)=ql                                             ! conLoad->Solutions
           conWidthB(j,i)=bl                                           ! conWidthB->Solutions
           conPress(j,i)=conLoad(j,i)/(pi*conWidthA(j,i)              & ! conPress->Solutions
     &                   *conWidthB(j,i))
           dqn(j)=conLoad(j,i)/conDef(j,i)                             ! dqn->LoadDer
        end if
        dqr(j)=dqn(j)*raceCosTaper(j)*sig(j)                           ! dqr->LoadDer, raceCosTaper->BrgGeom, sig->Constants
        dqx(j)=dqr(j)*raceTanTaper(j)                                  ! dqx->LoadDer, raceTanTaper->BrgGeom
        dqxx(j)=-dqx(j)                                                ! dqxx->LoadDer
        dqyy(j)=dqr(j)*ssi                                             ! dqyy->LoadDer, ssi from main part of Adrc1
        dqzz(j)=-dqr(j)*csi                                            ! dqzz->LoadDer, csi from main part of Adrc1
        qcl(j,i)=conLoad(j,i)                                          ! qcl->Solutions
        acl(j,i)=one3*four*conWidthA(j,i)                              ! acl->Solutions, one3,four->Constants
     else
        conLoad(j,i)=zero                                              ! conLoad,conWidthA,conWidthB,qcl,acl ->Solutions
        conWidthA(j,i)=zero                                            ! zero->Constants
        conWidthB(j,i)=zero
        qcl(j,i)=zero
        acl(j,i)=zero
        conPress(j,i)=zero
     end if

!
!       contact position
!       ----------------
!
        ccbgr=uvr*reCrn(i)+bcbgr                                       ! contact center relative to re geo center
        tbc=matmul(tac,tbia)                                           ! roll ele to contact frame transformation, tbc->ReVecs
        trc=matmul(tbc,transpose(tbr))                                 ! race to contact frame transformation, trc->ReVecs
        do k=1,3
           cbs(k,j,i)=zero
           crs(k,j,i)=zero
           do l=1,3                                                    ! cbs,crs->VecMats, both vectors in contact frame
              cbs(k,j,i)=cbs(k,j,i)+trc(k,l)*ccbgr(l)                 & ! contact position relative to re geo center
     &                   +tbc(k,l)*reGeoCen(l,i)                       ! add terms to locate cont pos rel to re mass center
              crs(k,j,i)=crs(k,j,i)+trc(k,l)*                         & ! contact position relative race mass center
     &                   (ccbgr(l)+bgrgr(l)+raceGeoCen(l,j))
           end do
        end do
        return
        end subroutine PointContact
        subroutine LineContact
!-----------------------------------------------------------------
!       line contact solutions
!-----------------------------------------------------------------
!
        integer                              :: i,j                    ! roller # and race #
        integer                              :: k,l,ll,lj              ! loop indices
        real(r8),dimension(2)                :: pqs                    ! elastic constants for line contact computation

        integer                              :: ksol                   ! flag for computing contact length
                                                                       ! 0 =previously computed coordinates are valid
                                                                       ! 1 compute new values
        integer                              :: jsol                   ! flag for computing local interaction
                                                                       ! 0 =no load condition, do not compute local interactions
        real(r8),dimension(2)                :: xj                     ! normalized x-coordinate defining the contact zone
        real(r8)                             :: aa                     ! contact length
        real(r8)                             :: xxo                    ! x-ccordinate of center of contact
        real(r8)                             :: qbs,qcs,bls,ccs,cc     ! integrated load,halfwith and geom interaction
        real(r8)                             :: qc,bl
        real(r8)                             :: ccm                    ! max geom interaction
        real(r8)                             :: xx,wk                  ! Gaussian abscissa and weight
        real(r8)                             :: ccj                    ! local geom interaction
        real(r8),dimension(3,3)              :: tira                   ! transformation from inertial to race azimuth frame
        real(r8)                             :: xtx,ytx,drx,dqxj       ! local variables for computing derivatives
        real(r8)                             :: dqtj,dqrj,dqnj
        real(r8)                             :: bmj,rmj                ! moment on roller and race at grid j
        real(r8)                             :: sum                    ! curvature sum
        real(r8)                             :: Adrc5                  ! geom interaction function
!
!       initialize solutions
!       --------------------
!
        i=ire                                                          ! ire->reVecs
        j=irace                                                        ! irace->reVecs
        bm=zero                                                        ! bm,rm->LoadDer
        rm=zero
        drx=-raceTanTaper(j)
        dqx(j)=zero
        dqr(j)=zero
        dqt(j)=zero                                                    ! all derivatives in LoadDer
        dqp(j)=zero
        dmx(j)=zero
        dmr(j)=zero
        dmt(j)=zero
        dqxx(j)=zero
        dqyy(j)=zero
        dqzz(j)=zero
        dqt2(j)=zero
```

```
        dqt3(j)=zero
        dmxx(j)=zero
        dmyy(j)=zero
        dmzz(j)=zero
        dmt2(j)=zero
        dmt3(j)=zero
        pqs(1)=qReRace(1,j)                                     ! qReRace->Constants
        pqs(2)=qReRace(2,j)
        tac(:,:)=tacs(:,:,j,i)                                  ! race azimuth to contact frame transformation
                                                                ! tac->ReVecs, tacs->VecMats
        qcs=zero                                                ! initialization for computation of averages
        qbs=zero
        bls=zero
        ccs=zero
!-------------------------------------
!       check existing contact limits
!       -----------------------------
!
        call OldContLmts(ksol,xj)                               ! check old contact limit coordinates
        if (ksol == 0) then
           xx=one2*(xj(1)+xj(2))                                ! check if contact exists
           ccj=Adrc5(xx)
           if (dabs(ccj) <= tolr(1)) then
              ccj=zero
              ccm=ccj
              jsol=0
           else if (ccj < zero) then
              ccm=ccj
              jsol=0
           else
              jsol=1
           end if
        else
           call NewContLmts(jsol,xj)                            ! compute new values when ksol /= 0
           ccm=conDef(j,i)                                      ! in case of no contact ccm = max cls
        end if
        conLmts(1,j,i)=xj(1)                                    ! store results for future use
        conLmts(2,j,i)=xj(2)                                    ! conLmts->Solutions
        if (jsol == 0) then                                     ! when jsol = 0, there is no contact
           qc=zero
           bl=zero
        else
!
!          local interactions in contact frame
!          -----------------------------------
!
           aa=(xj(2)-xj(1))/reCosTaper(i)                       ! reCosTaper->BrgGeom
           acl(j,i)=aa                                          ! acl->Solutions
           conWidthA(j,i)=one2*aa                               ! major contact halfwidth, conWidthA->Solutions
           aa=aa**expLen(j,iRaceRot(j))                         ! contact lenth parameter for life calculation
           xxo=one2*(xj(1)+xj(2))                               ! axial position of center of contact

           ccm=-1.0e+10_r8                                      ! nGrid,xgl,wgl->reVecs
           do k=1,nGrid                                           ! start loop for local interaction
              xx=conWidthA(j,i)*xgl(k)+xxo                      ! grid is in accordance with Gaussian quadrature
              wk=conWidthA(j,i)*wgl(k)
              ccj=Adrc5(xx)
              asc(1,k,j)=tra(2,2)                               ! store cosine of azimuth angle, asc->reVecs
              asc(2,k,j)=tra(2,3)                               ! store sine of azimuth angle
              do l=1,3
                 cbsl(l,k,j)=zero                               ! cbsl->reVecs
                 crsl(l,k,j)=zero                               ! crsl->reVecs
                 do ll=1,3                                      ! ccbgr,ccrgr->reVecs defined in Adrc5
                    cbsl(l,k,j)=cbsl(l,k,j)+trc(l,ll)*ccbgr(ll)      & ! contact center rel to roller center in contact frame
     &                          +tbc(l,ll)*reGeoCen(ll,i)       ! add term to locate relative to mass center
                    crsl(l,k,j)=crsl(l,k,j)+trc(l,ll)*(ccrgr(ll)     & ! contact center rel to race center in contact frame
     &                          +raceGeoCen(ll,j))              ! add vector to locate rel to mass center
                    tira(l,ll)=zero                             ! transformation from inertial to race azimuth frame
                    do lj=1,3
                       tira(l,ll)=tira(l,ll)+tra(l,lj)*tir(lj,ll)
                    end do
                 end do
              end do
!
!             elastic line contact solution
!             -----------------------------
!
              sum=(rRad-sig(j)*bRad)*raceCosTaper(j)/(bRad*rRad)     ! sig->constants,raceCosTaper->BrgGeom
              ccs=ccs+wgl(k)*ccj                                ! wgl->reVecs
              if(ccj > ccm) ccm=ccj
              call Adrc6(1,ccj,pqs,one,sum,bl,dqnj,qc)          ! one->Constants
              xCoord(k,j,i)=cbsl(1,k,j)                         ! store x-coordinate of grid point
              conDefDis(k,j,i)=ccj                              ! geometric interaction at grid point
              conLoadDis(k,j,i)=qc                              ! conLoadDis->Solutions
              conWidthDis(k,j,i)=bl                             ! conWidthDis->Solutions
              qcs=qcs+wk*qc**expLoadA(j,iRaceRot(j))            ! expLoadA,iRaceRot->fatigueCons
              dqn(j)=dqnj
              bls=bls+wgl(k)*bl
              qbs=qbs+wk*qc
              call LCDerivatives(qc,dqnj,bmj,rmj)
              bm(j)=bm(j)+wk*bmj                                ! moment on roller, bm->LoadDer
              rm(j)=rm(j)+wk*rmj                                ! moment on race, rm->LoadDer
```

```
            if (ider /= 0) then                                    ! ider->LoadDer, set derivatives
               dqx(j)=dqx(j)+wk*delQbj(1)                          ! dqx->LoadDer
               dqr(j)=dqr(j)+wk*delQbj(2)                          ! dqr->LoadDer
               dqt(j)=dqt(j)+wk*delQbj(3)                          ! dqt->LoadDer
               dqp(j)=dqp(j)+wk*delQbj(4)                          ! dqp->LoadDer
               dmx(j)=dmx(j)+wk*delMbj(1)                          ! dmx->LoadDer
               dmr(j)=dmr(j)+wk*delMbj(2)                          ! dmr->LoadDer
               dmt(j)=dmt(j)+wk*delMbj(3)                          ! dmt->LoadDer
               dmp(j)=dmt(j)+wk*delMbj(4)                          ! dmp->LoadDer
               dqxx(j)=dqxx(j)+wk*delQrj(1)                        ! dqxx->LoadDer
               dqyy(j)=dqyy(j)+wk*delQrj(2)                        ! dqyy->LoadDer
               dqzz(j)=dqzz(j)+wk*delQrj(3)                        ! dqzz->LoadDer
               dqt2(j)=dqt2(j)+wk*delQrj(4)                        ! dqt2->LoadDer
               dqt3(j)=dqt3(j)+wk*delQrj(5)                        ! dqt3->LoadDer
               dmxx(j)=dmxx(j)+wk*delMrj(1)                        ! dmxx->LoadDer
               dmyy(j)=dmyy(j)+wk*delMrj(2)                        ! dmyy->LoadDer
               dmzz(j)=dmzz(j)+wk*delMrj(3)                        ! dmzz->LoadDer
               dmt2(j)=dmt2(j)+wk*delMrj(4)                        ! dmt2->LoadDer
               dmt3(j)=dmt3(j)+wk*delMrj(5)                        ! dmt3->LoadDer
            end if
         end do
         qc=aa*qcs**expLoadAI(j,iRaceRot(j))
         cc=one2*ccs
         bls=one2*bls
         xx=one2*(xj(1)+xj(2))
!
!        parameters at center of contact
!        -------------------------------
!
         ccj=Adrc5(xx)
         asc(1,ngm,j)=tra(2,2)
         asc(2,ngm,j)=tra(2,3)
         sum=(rRad-sig(j)*bRad)*raceCosTaper(j)/(bRad*rRad)
         radX(j)=one/sum                                           ! store equivalent radius for trac procs, radX->ReVecs
         call Adrc6(1,ccj,pqs,one,sum,bl,dqnj,qc)
         do l=1,3
            cbsl(l,ngm,j)=zero
            crsl(l,ngm,j)=zero
            do ll=1,3                                              ! trc,tbc,ccbgr,ccrgr->ReVecs, set in Adrc5
               cbsl(l,ngm,j)=cbsl(l,ngm,j)+trc(l,ll)*ccbgr(ll)     & ! contact center rel to roller center in contact frame
    &                       +tbc(l,ll)*reGeoCen(ll,i)              ! add term to locate relative to mass center
               crsl(l,ngm,j)=crsl(l,ngm,j)+trc(l,ll)*(ccrgr(ll)    & ! contact center rel to race center in contact frame
    &                     +raceGeoCen(ll,j))                       ! add vector to locate rel to mass center
            end do
         end do
         do l=1,3
            cbs(l,j,i)=cbsl(l,ngm,j)
            crs(l,j,i)=crsl(l,ngm,j)
         end do
      end if
      conLoadDis(ngm,j,i)=qc                                       ! conLoadDis->Solutions
      conWidthDis(ngm,j,i)=bl                                      ! conWidthDis->Solutions
      conDef(j,i)=ccm                                              ! conDef->Solutions
      conLoad(j,i)=qbs                                             ! conLoad->Solutions
      conWidthB(j,i)=bls                                           ! conWidthB->Solutions
      qcl(j,i)=qc                                                  ! qcl->Solutions
      if (qc > zero) then
         conPress(j,i)=qc/(pi*bls*conWidthA(j,i))                  ! conPress->Solutions
      end if
      return
      end subroutine LineContact
      subroutine OldContLmts(ksol,xj)
!------------------------------------------------------------
!     contact limit coordinates from existing solutions
!------------------------------------------------------------
      integer                              :: i,j                  ! re and race #
      integer                              :: ksol                 ! 0 =computations based on existing solutions are ok
                                                                   ! 1 =new solutions are requid
      real(r8),dimension(2)                :: xj                   ! computed contact limits
      real(r8)                             :: xab,xa,xaa,x1,x2     ! local variables for contact length
      real(r8)                             :: cci,ccj              ! local variables for geometric interaction
      integer                              :: k                    ! loop index
      integer                              :: ier                  ! error indicator
      real(r8)                             :: Adrc5                ! geometric interaction function
      external Adrc5
      i=ire
      j=irace
      xab=one2*dabs(conLmts(2,j,i)-conLmts(1,j,i))                 ! contact length from earlier solutions
      ksol=0                                                       ! conLmts->Solutions
      k=1
      do while (k <= 2)
         xa=conLmts(k,j,i)
         ccj=Adrc5(xa)
         if (dabs(ccj) < tolr(1)) then                             ! stored solution is valid, tolr->Constants
            xj(k)=xa
         else if(ccj > zero .and.                                  &
    &       dabs(xa-reConLmt(k,i)) < tolr(1)) then                 ! contact exists for entire length of roller
            xj(k)=xa                                               ! reConLmt->BrgGeom
         else                                                      ! modify existing contact coordinate
            xaa=one10*dabs(reConLmt(k,i))                          ! length increment for updating contact length
            if (xaa > xab) xaa=xab                                 ! reConLmt->BrgGeom
            x1=xa-xaa
```

```
            x2=xa+xaa
            if(k == 1) then                                        ! check coordinate against max permissible con length
               if(x1 < reConLmt(k,i)) x1=reConLmt(k,i)             ! reConLmt->BrgGeom
            else
               if(x2 > reConLmt(k,i)) x2=reConLmt(k,i)             ! reConLmt->BrgGeom
            end if
            call Adrg3(xa,x1,x2,Adrc5,ier)                         ! solve for new coordinate value with old
            if(ier == 0) then                                      ! value as a first guess
               xj(k)=xa
            else
               ksol=1                                              ! set switch for more generalized computations when
               k=2                                                 ! a solution near the old value cannot be found
            end if
         end if
         if(k == 1 .and. ksol == 0) then                           ! look for symetrical solutions
            cci=Adrc5(xa)
            xa=-xa
            ccj=Adrc5(xa)
            if(dabs(cci-ccj) < tolr(1)) then
               xj(2)=xa
               k=2
            else if(ccj > zero .and.                               & ! check for limiting coordinate
     &         dabs(xa-reConLmt(k,i)) <= tolr(1)) then             ! reConLmt->BrgGeom
               xj(2)=xa
               k=2
            end if
         end if
         k=k+1
      end do
      return
      end subroutine OldContLmts
      subroutine NewContLmts(jsol,xj)
!-------------------------------------------------------------
!     compute new coordinates for defining contact length
!-------------------------------------------------------------
      integer                                :: i,j                ! roller and race #
      integer                                :: jsol               ! contact solution flag
                                                                   ! 0 =no contact, 1 =contact exists
      real(r8),dimension(2)                  :: xj                 ! computed contact limit coordinate values
                                                                   ! array also used for lower bound for iterative soln
      integer                                :: nx =5              ! number of grid points for coarse bounds on contact length
      real(r8),dimension(5)                  :: xxi = (/1.0_r8,    & ! grid points for coarse bounds
     &                            0.75_r8,0.50_r8,0.25_r8,0.0_r8/)
      real(r8)                               :: cc,ccm,cci,ccj     ! local variables for geom interaction
      integer                                :: k,l                ! loop index
      integer,dimension(2)                   :: jl                 ! grid indices for contact limit values
      integer                                :: ll,lk              ! grid indices used for contact limits
      real(r8),dimension(2)                  :: xjk                ! array used for upper bound for iterative soln
      real(r8)                               :: xa                 ! local coordinate along roller axis
      integer                                :: ier                ! error indicator
      real(r8)                               :: Adrc5              ! geometric interaction function
      external  Adrc5
      i=ire
      j=irace
!
!     maximum geom interaction and grid indices for contact limit
!     ---------------------------------------------------------
!
      jl=0
      ccm=-1.0e+10_r8
      k=1
      do while (k <= nx)
         do l=1,2
            if (jl(l) == 0) then
               xa=xxi(k)*reConLmt(l,i)                             ! reConLmt->BrgGeom
               cc=Adrc5(xa)
               if (cc > ccm) ccm=cc                                ! maximum interaction
               if (cc > tolr(1)) then                              ! check for positive interaction, tolr->Constants
                  jl(l)=k
                  xj(l)=xa
                  if (k > 1) xjk(l)=xxi(k-1)*reConLmt(l,i)         ! if k=1, then xj is the desired coordinate value
               end if                                              ! and no iterative computations will be requid
            end if                                                 ! reConLmt->BrgGeom
         end do
         if(jl(1) /= 0 .and. jl(2) /= 0) k=nx+1                    ! terminate loop when both limits are found
         k=k+1
      end do
      jsol=1                                                       ! set initial value to indicate contact
      if (jl(1) == 0 .and. jl(2)  == 0) then
!-------------------------------
!        no contact condition
!        --------------------
         conWidthA(j,i)=zero                                       ! conWidthA->Solutions, zero->Constants
         conWidthB(j,i)=zero                                       ! conWidthB->Solutions
         conDef(j,i)=ccm                                           ! conDef->Solutions
         conLoad(j,i)=zero                                         ! conLoad->Soluitons
         conLmts(1,j,i)=reConLmt(1,i)                              ! conLmts->Solutions
         conLmts(2,j,i)=reConLmt(2,i)                              ! conLmts->Solutions
         xj(1)=reConLmt(1,i)
         xj(2)=reConLmt(2,i)
         jsol=0                                                    ! contact flag for no contact
         return
```

```
        else if (jl(1)*jl(2) == 0) then
!---------------------------------------------------
!          partial contact at either end of roller
!          contact limit coordinate defines one
!          end of the contact zone
!          ----------------------------------------
           if (jl(1) > 0) then
              ll=2                                          ! index of contact limit point to be computed
              lk=1                                          ! index of point defined by contact limit
           else
              ll=1                                          ! index of contact limit point to be computed
              lk=2                                          ! index of point defined by contact limit
           end if
           k=4                                              ! determine point of positive interaction for the
           do while (k > 1)                                 ! limiting point defined by index ll
              xa=xxi(k)*reConLmt(lk,i)                      ! scan backwards starting with grid point #4 (0.25)
              cc=Adrc5(xa)
              if (cc > tolr(1)) then
                 jl(ll)=k
                 xj(ll)=xa
                 xjk(ll)=xxi(k+1)*reConLmt(lk,i)
                 k=0
              end if
              k=k-1
           end do
           if (k == 1) then                                 ! when contact in within the first grid point
              jl(ll)=2                                      ! near edge of roller
              xj(ll)=xxi(2)*reConLmt(lk,i)
              xjk(ll)=xxi(1)*reConLmt(lk,i)
           end if
        end if
!
!       compute axial coordinates for contact length
!       --------------------------------------------
!
        k=1
        do while (k <= 2)
           if(jl(k) /= 1) then                             ! when this index corresponds to first grid point
              if(xjk(k) < xj(k)) then                      ! the value in xj already contains the desid soln
                 xa=xjk(k)
                 xjk(k)=xj(k)                              ! xj and xjk are sorted in ascending order
                 xj(k)=xa
              end if
              xa=one2*(xjk(k)+xj(k))                       ! initial guess is at mid point
              call Adrg3(xa,xj(k),xjk(k),Adrc5,ier)        ! compute iterative solution
              if(ier /= 0) then
                 ierInd=kRootLC                            ! kRootLC->ErrorCodes, ierInd->Errors
                 jerInd=ier                                ! jerInd->Errors
                 iMess(1)='Error in LineContact procedure in Adrc1'   ! iMess->Errors
                 numM=1                                    ! numM->Errors
                 return
              end if
              xj(k)=xa
              if(k.eq.1) then
                 cci=Adrc5(xa)
                 xa=-xa                                    ! look for symetrical solution
                 ccj=Adrc5(xa)
                 if (dabs(cci-ccj) < tolr(1)) then
                    xj(2)=xa
                    k=2
                 end if
              end if
           end if
           k=k+1
        end do
        return
        end subroutine NewContLmts
        subroutine LCDerivatives(qj,sj,bmj,rmj)
!---------------------------------------------------------------------
!       line contact load and moment derivaties at grid point j
!       all derivative terms are set in variables declared in Adrc1
!---------------------------------------------------------------------

        real(r8)                        :: qj                ! input load at grid point
        real(r8)                        :: sj                ! stiffness at grid point
        real(r8)                        :: bmj,rmj           ! moment on roller and race about y axis in
                                                             ! in contact frame
        real(r8)                        :: delXX             ! derivative of deflection with race position X
        real(r8)                        :: delYY             ! derivative of deflection with race position Y
        real(r8)                        :: delZZ             ! derivative of deflection with race position Z
        real(r8)                        :: delx              ! derivative of deflection with roller position X
        real(r8)                        :: delr              ! derivative of deflection with roller position R
        real(r8)                        :: delAng2r          ! derivative of deflection with race trans ang 2
        real(r8)                        :: delAng3r          ! derivative of deflection with race trans ang 3
        real(r8)                        :: delAng2b          ! derivative of deflection with roller trans ang 2
        real(r8)                        :: delAng3b          ! derivative of deflection with roller trans ang 3
        real(r8)                        :: armXX             ! derivative of moment arm with race position X
        real(r8)                        :: armYY             ! derivative of moment arm with race position Y
        real(r8)                        :: armZZ             ! derivative of moment arm with race position Z
        real(r8)                        :: armx              ! derivative of moment arm with roller position X
        real(r8)                        :: armr              ! derivative of moment arm with roller position R
        real(r8)                        :: armAng2r          ! derivative of moment arm with race trans ang 2
```

```
      real(r8)                                 :: armAng3r            ! derivative of moment arm with race trans ang 3
      real(r8)                                 :: armAng2b            ! derivative of moment arm with roller trans ang 2
      real(r8)                                 :: armAng3b            ! derivative of moment arm with roller trans ang 3
      real(r8),dimension(3,3)                  :: tira                ! inertial to race azimuth transformation
      real(r8),dimension(3,3)                  :: t1,t2,t3,t4         ! local transformation matrix
      real(r8),dimension(3)                    :: r1,r2               ! local vectors
      real(r8)                                 :: arm1,arm2           ! moment arm on race and roller in contact frame
      tira=matmul(tra,tir)                                            ! tra,tir->ReVecs, inertial to race azimuth
      r2=matmul(transpose(tib),ccbgb)                                 ! ccbgb->ReVecs, roller cont point -inertial frame
      r1=r2+bgrgi                                                     ! bgrgi->Adrc1, race cont point - iertial frame
      t4=matmul(tac,tira)                                             ! inertial to contact transformation
      arm1=t4(1,1)*r1(1)+t4(1,2)*r1(2)+t4(1,3)*r1(3)                  ! moment arm for race
      arm2=t4(1,1)*r2(1)+t4(1,2)*r2(2)+t4(1,3)*r2(3)                  ! moment arm for roller
      bmj=qj*arm2                                                     ! moment on roller
      rmj=-qj*arm1                                                    ! moment on race
      if (ider == 0) return                                           ! return if no derivatives are required
!------------------------------------------------------------         ! ider->LoadDer
!        derivatives of deflection and moment arm for race
!        -------------------------------------------------
      delXX=-tira(3,1)*tac(1,1)                                       ! derivative of def with race dis X
                                                                      ! tac->ReVecs  tac(1,1)=cos(alfa)
      delYY=-tira(3,2)*tac(1,1)                                       ! derivative of def with race dis Y
      delZZ=-tira(3,3)*tac(1,1)                                       ! derivative of def with race dis Z
      armXX=-t4(1,1)                                                  ! moment arm with race dis X
      armYY=-t4(1,2)                                                  ! moment arm with race dis Y
      armZZ=-t4(1,3)                                                  ! moment arm with race dis Z
      t1=matmul(tra,tir2)                                             ! tra->ReVecs tir2->Adrc1
      t2=matmul(tra,tir3)                                             ! tir3->Adrc1
      t3=matmul(tac,t1)
      t4=matmul(tac,t2)
      delAng2r=zero                                                   ! derivative of del with race angle 2
      delAng3r=zero                                                   ! derivative of del with race angle 3
      armAng2r=zero                                                   ! derivative of moment arm with race angle 2
      armAng3r=zero                                                   ! derivative of moment arm with race angle 3
      do k=1,3
         delAng2r=delAng2r+t1(3,k)*r1(k)*tac(1,1)
         delAng3r=delAng3r+t2(3,k)*r1(k)*tac(1,1)
         armAng2r=armAng2r+t3(1,k)*r1(k)
         armAng3r=armAng3r+t4(1,k)*r1(k)
      end do
!--------------------------------------------------------------
!        derivatives of deflection and moment arm for roller
!        ---------------------------------------------------
      delx=tira(3,1)*tac(1,1)                                         ! derivative of def with roller dis x
      delr=(-tira(3,2)*sbsi+tira(3,3)*cbsi)*tac(1,1)                  ! derivative of def with roller dis r
                                                                      ! sbsi,cbsi->Adrc1, defined in Adrc5
      t1=matmul(tira,transpose(tib2))                                 ! tib2,tib3->Adrc1
      t2=matmul(tira,transpose(tib3))
      t3=matmul(tac,t1)
      t4=matmul(tac,t2)
      delAng2b=zero                                                   ! derivative of del with roller angle 2
      delAng3b=zero                                                   ! derivative of del with roller angle 3
      armAng2b=zero                                                   ! derivative of moment arm with roller angle 2
      armAng3b=zero                                                   ! derivative of moment arm with roller angle 3
      do k=1,3
         delAng2b=delAng2b+t1(3,k)*ccbgb(k)*tac(1,1)                  ! ccbgb->ReVecs
         delAng3b=delAng3b+t2(3,k)*ccbgb(k)*tac(1,1)
         armAng2b=armAng2b+t3(1,k)*ccbgb(k)
         armAng3b=armAng3b+t4(1,k)*ccbgb(k)
      end do
!-------------------------------------------------------------------
!        assemble terms for load and moment derivaties for roller
!        ----------------------------------------------------------
      armx=zero                                                       ! derivative of moment arm with roller dis x
      armr=zero                                                       ! derivative of moment arm with roller dis r
      delQbj(1)=sj*delx                                               ! roller load der with displacement x
      delQbj(2)=sj*delr                                               ! roller load der with displacement r
      delQbj(3)=sj*delAng2b                                           ! roller load der with trans angle 2
      delQbj(4)=sj*delAng3b                                           ! roller load der with trans angle 3
      delMbj(1)=arm2*delQbj(1)+qj*armx                                ! roller moment der with disp x
      delMbj(2)=arm2*delQbj(2)+qj*armr                                ! roller moment der with disp r
      delMbj(3)=arm2*delQbj(3)+qj*armAng2b                            ! roller moment der with trans ang 2
      delMbj(4)=arm2*delQbj(4)+qj*armAng3b                            ! roller moment der with trans ang 3
!-----------------------------------------------------------------
!        assemble terms for load and moment derivaties for race
!        --------------------------------------------------------
      delQrj(1)=sj*delXX                                              ! race load der with disp x
      delQrj(2)=sj*delYY                                              ! race load der with disp y
      delQrj(3)=sj*delZZ                                              ! race load der with disp z
      delQrj(4)=sj*delAng2r                                           ! race load der with trans ang 2
      delQrj(5)=sj*delAng3r                                           ! race load der with trans ang 3
      delMrj(1)=-arm1*delQrj(1)-qj*armXX                              ! race moment der with disp x
      delMrj(2)=-arm1*delQrj(2)-qj*armYY                              ! race moment der with disp y
      delMrj(3)=-arm1*delQrj(3)-qj*armZZ                              ! race moment der with disp z
      delMrj(4)=-arm1*delQrj(4)-qj*armAng2r                           ! race moment der with trans ang 2
      delMrj(5)=-arm1*delQrj(5)-qj*armAng3r                           ! race moment der with trans ang 3
      return
      end subroutine LCDerivatives
      subroutine FlngContact
!--------------------------------------------------------------------
!        flange contacts for roller bearings
!--------------------------------------------------------------------
```

```
        integer                             :: i,j                 ! roller and race #
        integer                             :: k,l,m,ll            ! loop indices
        integer                             :: ltc                 ! flange contact code
        i=ire
        j=irace
!-------------------------
!       set up velocities
!       -----------------
        do k=1,3
           vbr(k)=zero                                             ! roller velocity, vbr->FlngData, zero->Constants
           ombr(k)=zero                                            ! roller ang velocity, ombr->FlngData
           vrr(k)=zero                                             ! race velocity, vrr->FlngData
           omrr(k)=raceAngVel(k,irace)                             ! race angular velocity, omrr->FlngData
           do l=1,3
              vbr(k)=vbr(k)+tir(k,l)*reVel(l,ire)                  ! transform all velocities to race frame
              ombr(k)=ombr(k)+tbr(k,l)*reAngVel(l,ire)
              vrr(k)=vrr(k)+tir(k,l)*raceVel(l,irace)
           end do
        end do
!------------------------------
!       start loop for flanges
!       ----------------------
        do k=1,2
           if (kFlngInd(k,irace) /= 0) then                        ! kFlngInd->BrgGeom
!------------------------
!             setup data
!             ----------
              iArray(2)=k                                          ! store flange indices in case of error
              iArray(3)=j                                          ! iArray->Errors
              iArray(1)=i
              raceFlngWRi(k,j,i)=zero                              ! initialize solutions
              flngSV(k,j,i)=zero                                   ! raceFlngWRi,flngSV->Solutions
              xcc=reCorCen(1,k,i)                                  ! xcc->FlngData
              rcc=reCorCen(2,k,i)                                  ! rcc->FlngData
              do l=1,3
                 do m=1,3                                          ! ter->FlngData
                    ter(l,m)=zero
                    do ll=1,3
                       ter(l,m)=ter(l,m)+tbr(l,ll)*tre(m,ll,k,i)   ! tre->BrgGeom
                    end do
                 end do
              end do
              rf(1)=flngOrigin(1,k,j)                              ! rf->flngData
              rf(2)=zero
              rf(3)=flngOrigin(2,k,j)+raceExp(j)                   ! raceExp->OpCond
              trafjk=traf(:,:,k,j)                                 ! trafjk->FlngData, traf->BrgGeom
              trf=matmul(trafjk,tra)                               ! trf->FlngData, tra->ReVecs
              tria=raceToIaz(:,:,k,i)                              ! tria->ReVecs
              eRad=reEndRad(k,i)                                   ! eRad->FlngData
              corRad=reCorRad(k,i)                                 ! corRad-> roller corner radius
              ecc=reEndCen(k,i)                                    ! ecc->FlngData
              fHt=flngHt(k,j)                                      ! fHt->FlngData
              conf(:)=flngCons(:,k,j)                              ! conf->FlngData, flngCons->Constants
              if (iStat(k,j,i) > 0) then                           ! iStat->FlngData
                 th=ths(k,j,i)                                     ! angular coordinate for cor int
              else                                                 ! th,ths,reRotc,reRotp->FlngData
                 th=ths(k,j,i)-reRotc(i)+reRotp(i)
              end if
              call Adrc3(iStat(k,j,i),ltc)                         ! flange interaction procedure
              if (ierInd /= 0) call Adra9                          ! process error if any
!---------------------------
!             store results
!             -------------
              iStat(k,j,i)=1                                       ! iStat->FlngData
              idFlngCon(k,j,i)=ltc                                 ! idFlngCon->Solutions
              flngForce(k,j,i)=qf                                  ! flngForce->Solutions
              fqdel(k,j)=qdel                                      ! store contact stiffness, fqdel-LoadDer
                                                                   ! qdel->FlngData
              if (qf > zero) idFlngSol=idFlngSol+1                 ! idFlngSol->ReVecs
              flngConWidthA(k,j,i)=af                              ! flngConWidthA->Solutions
              flngConWidthB(k,j,i)=bf                              ! flngConWidthB->Solutions
              flngDef(k,j,i)=delf                                  ! flngDef->Solutions
              flngConPos(k,j,i)=ccff(3)                            ! ccff->flngData
              reEndWR(k,i)=reEndWR(k,i)+qvf                        ! reEndWR->Solutions
              raceFlngWRi(k,j,i)=qvf                               ! raceFlngWRi->solutions
              ths(k,j,i)=th                                        ! ths,th->FlngData
              flngSV(k,j,i)=svf                                    ! flngSV->Solutions
              do l=1,3
                 raceFlngFor(l,j)=raceFlngFor(l,j)+raceFlngForjk(l)    ! raceFlngFor->ReVecs
                 raceFlngMom(l,j)=raceFlngMom(l,j)+raceFlngMomjk(l)    ! raceFlngMom->ReVecs
                 reFlngFor(l)=reFlngFor(l)+reFlngForjk(l)              ! reFlngFor->ReVecs
                 reFlngMom(l)=reFlngMom(l)+reFlngMomjk(l)              ! reFlngMom->ReVecs
                 rFlngFor(l,j)=rFlngFor(l,j)+rFlngForjk(l)             ! rFlngFor->ReVecs
                 rFlngMom(l,j)=rFlngMom(l,j)+rFlngMomjk(l)             ! rFlngMom->ReVecs
                 bFlngFor(l)=bFlngFor(l)+bFlngForjk(l)                 ! bFlngFor->ReVecs
                 bFlngMom(l)=bFlngMom(l)+bFlngMomjk(l)
                 do m=1,4                                              ! transform der and unit vectors to azimuth frame
                    dbFlngFor(l,m)=dbFlngFor(l,m)+dbFlngForjk(l,m)     ! derivative of roller flange force
                    dbFlngMom(l,m)=dbFlngMom(l,m)+dbFlngForjk(l,m)     ! derivative of roller flange moment
                    drFlngFor(l,m,j)=drFlngFor(l,m,j)+drFlngForjk(l,m) ! derivative of race flange force (comp 1-4)
                    drFlngMom(l,m,j)=drFlngMom(l,m,j)+drFlngMomjk(l,m) ! derivative of race flange moment (comp 1-4)
                 end do
```

```
            drFlngFor(l,5,j)=drFlngFor(l,5,j)+drFlngForjk(l,5)    ! derivative of race flange force (comp 5)
            drFlngMom(l,5,j)=drFlngMom(l,5,j)+drFlngMomjk(l,5)    ! derivative of race flange moment (comp 5)
         end do
      end if
   end do
   return
   end subroutine FlngContact
end subroutine Adrc1
```

Appendix C - Ball Bearing Data

Adore input data for the ball bearing case with the associated print ouput is documented below. Only the output and initial and final time steps is included for brevity.

```
Listing of input data records
-----------------------------


Rec 1            1     0     0     0     0     1   500     1     0     0     1  2500     0     5
Rec 2.1      5.0000E-02 5.0000E-03 3.0000E-01 1.0000E+03 1.0000E-04 0.0000E+00
Rec 2.4          1     0     0     2
Rec 2.5      3.2300E+02 1.0000E-05 0.0000E+00 3.0000E-03 3.0000E-02
Rec 2.7      3.2300E+02 3.2300E+02 3.2300E+02 3.2300E+02 3.2300E+02 3.2300E+02
Rec 3.1     Ball Bearing Test Case kGeoMod=100
Rec 3.2       1  1 18  1  0  0  0  0  0  0  0  0
Rec 3.3       0  1  0  1  1  0  0  0  0  0  0  0  0  0  1
Rec 3.4       0  0  0  0  0  2  1 19  0  0  0  0
Rec 4        1.0000E-01 1.8000E-01 2.0000E-02 2.0500E-01 3.0000E-02 3.0000E-02
Rec 5A       1.9050E-02 1.4000E-01 2.5000E+01 0.0000E+00 5.2000E-01 5.4000E-01
Rec 7.0       0  0  0  0  0  0  0  0  2  2  2
Rec 7.1      1.4629E-01 1.3007E-01 3.6241E-02 5.0000E-03 1.5000E-03 1.2000E-03 0.0000E+00 0.0000E+00 0.0000E+00
Rec 7.2.1    1.3007E-01 8.0000E-03 1.8120E-02 1.5000E-03
Rec 7.2.2    1.3007E-01 8.0000E-03 1.8120E-02 1.5000E-03
Rec 9.0      3.2300E+02 1.0000E-05 5.0000E-05
Rec 9.1.1    4.5000E+03 0.0000E+00 2.0000E+03 0.0000E+00 0.0000E+00 0.0000E+00
Rec 9.1.2    0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 2.0000E+04
Rec 9.5.1    6.6360E-04 1.0000E+00 1.0000E+00 0.0000E+00 0.0000E+00-6.0000E-04 0.0000E+00 0.0000E+00 0.0000E+00
Rec 10.0      3  1  0  0  0
Rec 10.1C    0.0000E+00 2.0000E-02 1.6000E-02 1.0000E+00
Rec 10.3     1.2500E-07 1.0000E+01
Rec 10.5.1B  5.0000E-02 0.0000E+00 0.0000E+00 0.0000E+00 1.2500E-07
Rec 10.5.2B  5.0000E-02 0.0000E+00 0.0000E+00 0.0000E+00 1.2500E-07
Rec 11       0.0000E+00 0.0000E+00-9.8100E+00


Input from optional user subroutines
------------------------------------
```

```
                                              ADVANCED DYNAMICS OF ROLLING ELEMENTS
                                              =====================================
                                        A Real Time Simulation of Rolling Bearing Performance
                                                      [ Version (ADORE 5.0) ]


                                                            Copyright
                                                       Pradeep K Gupta Inc

  ****************************************************************************************************************************
  *                                                                                                                          *
  *     Bearing Type = BALL                            Program Mode =  1       Run Id = Ball Bearing Test Case kGeoMod=100     *
  ****************************************************************************************************************************


  BEARING GEOMETRY
  ----------------

  NO OF BALLS      18            OUTER CUR FAC   5.20000E-01     OUTER FIT   (m) 1.00000E-05     SHAFT OD    (m) 1.00000E-01
  BALL DIA    (m) 1.90500E-02    INNER CUR FAC   5.40000E-01     INNER FIT   (m) 5.00000E-05     SHAFT ID    (m) 2.00000E-02
  PITCH DIA   (m) 1.40000E-01    DIA PLAY    (m) 2.14180E-04     WIDTH  I    (m) 3.00000E-02     BEARING OD  (m) 1.80000E-01
  CON ANGLE (deg) 2.50000E+01    END PLAY    (m) 9.66105E-04     WIDTH II    (m) 3.00000E-02     HOUSING OD  (m) 2.05000E-01


  CAGE OD     (m) 1.46292E-01    CAGE WIDTH  (m) 3.62410E-02     POC CLS  I  (m) 1.20000E-03     OUTER CLS   (m) 5.00000E-03
  CAGE ID     (m) 1.30073E-01    SEG CUT   (deg) 0.00000E+00     POC CLS II  (m) 0.00000E+00     INNER CLS   (m) 1.50000E-03
  CAGE CONE (deg) 0.00000E+00    POCKET SHAPE     0              NO OF C/R LANDS  2              NO OF CAGE SEGS  1

  GUIDE  GUIDE     GUIDE LAND     GUIDE LAND     GUIDE LAND     GUIDE LAND
   LAND   TYPE            DIA            CLS          WIDTH       POSITION
                          (m)            (m)            (m)            (m)
     1      2    1.30073E-01    1.50000E-03    8.00000E-03    1.81204E-02
     2      2    1.30073E-01    1.50000E-03    8.00000E-03    1.81204E-02
```

MATERIAL PROPERTIES

		ROLLING ELEMENT	OUTER RACE	INNER RACE	CAGE	SHAFT	HOUSING
DENSITY	(kg/m**3)	7.75000E+03	7.75000E+03	7.75000E+03	7.75000E+03	7.75000E+03	7.75000E+03
ELASTIC MODULUS	(Pa)	2.00000E+11	2.00000E+11	2.00000E+11	2.00000E+11	2.00000E+11	2.00000E+11
POISSON-S RATIO		2.50000E-01	2.50000E-01	2.50000E-01	2.50000E-01	2.50000E-01	2.50000E-01
COEFF OF THERMAL EXP	(m/m/K)	1.17000E-05	1.17000E-05	1.17000E-05	1.80000E-05	1.17000E-05	1.17000E-05
THERMAL CONDUCTIVITY	(W/m/K)	4.30000E+01	4.30000E+01	4.30000E+01	4.30000E+01	4.30000E+01	4.30000E+01
HEAT CAPACITY	(J/kg/K)	4.70000E+02	4.70000E+02	4.70000E+02	4.70000E+02		
ELASTIC STRAIN LIMIT		2.00000E-03	2.00000E-03	2.00000E-03	2.00000E-03		
HARDNESS	(RC)	6.10000E+01	6.10000E+01	6.10000E+01	2.00000E+01		
WEAR COEFFICIENT		5.00000E-06	5.00000E-06	5.00000E-06	5.00000E-04		

INERTIAL PARAMETERS

	.......MASS (kg)	..........MOMENT OF INERTIA.......... (kg*m**2) X-COMP	Y-COMP	Z-COMP	.........MASS TO GEO CENTER.......... (m) X-COMP	Y-COMP	Z-COMP
RE	2.80534E-02	1.01806E-06	1.01806E-06	1.01806E-06	0.00000E+00	0.00000E+00	0.00000E+00
CAGE	6.24437E-01	2.99108E-03	1.56388E-03	1.56388E-03	0.00000E+00	0.00000E+00	0.00000E+00
OUTER RACE	1.29291E+00	9.32831E-03	4.76112E-03	4.76112E-03	0.00000E+00	0.00000E+00	0.00000E+00
INNER RACE	8.38958E-01	2.57921E-03	1.35252E-03	1.35252E-03	0.00000E+00	0.00000E+00	0.00000E+00

	.........PRINCIPAL TO GEO FRAME........ (deg) X-COMP	Y-COMP	Z-COMP
RE	0.00000E+00	0.00000E+00	0.00000E+00
CAGE	0.00000E+00	0.00000E+00	0.00000E+00
OUTER RACE	0.00000E+00	0.00000E+00	0.00000E+00
INNER RACE	0.00000E+00	0.00000E+00	0.00000E+00

LUBRICATION PARAMETERS

HYPOTHETICAL MODELS	CRITICAL FILM (m)	TRAC COEFF AT ZERO SLIP	MAXIMUM TRAC COEFF	TRAC COEFF AT INF SLIP	SLIP AT MAX TRACTION (m/s)	COEFFICIENT A	COEFFICIENT B (s/m)	COEFFICIENT C (s/m)	COEFFICIENT D
RE/RACE1	1.25000E-07	0.00000E+00	2.00000E-02	1.60000E-02	1.00000E+00	-1.60000E-02	3.82896E-02	1.71782E+00	1.60000E-02
RE/RACE2	1.25000E-07	0.00000E+00	2.00000E-02	1.60000E-02	1.00000E+00	-1.60000E-02	3.82896E-02	1.71782E+00	1.60000E-02
RE/CAGE	1.25000E-07				0.00000E+00	5.00000E-02	0.00000E+00	0.00000E+00	
RACE/CAGE	1.25000E-07				0.00000E+00	5.00000E-02	0.00000E+00	0.00000E+00	

LUBRICANT NAME	MIL-L-7808				
NOMINAL REF VISCOSITY	8.61565E-03 (Pa.s)	REF TEMP	3.23000E+02 (K)	THER COND	1.42518E-01 (W/m/K)
		DENSITY	9.45429E+02 (kg/m**3)	HEAT CAP	1.83610E+03 (J/kg/K)
PRESSURE COEFFICIENTS:	1.07562E-08 (1/Pa)		0.00000E+00 (1/Pa) **2		0.00000E+00 (1/Pa) **3
TEMP COEFFICIENTS:	2.96433E+03 (K)		0.00000E+00 (K)**2		0.00000E+00 (K)**3
PR-TEMP COEFFICIENTS:	0.00000E+00 (K/Pa)		0.00000E+00 (K/Pa) **2		0.00000E+00 (K/Pa) **3
TRACTION REF VISCOSITY	8.15882E-02 (Pa.s)				
PRESSURE COEFFICIENTS:	5.22140E-09 (1/Pa)		0.00000E+00 (1/Pa) **2		0.00000E+00 (1/Pa) **3
TEMP COEFFICIENTS:	4.25519E-02 (1/K)		0.00000E+00 (1/K)**2		0.00000E+00 (1/K)**3
PR-TEMP COEFFICIENTS:	0.00000E+00 (1/Pa/K)		0.00000E+00 (1/Pa/K) **2		0.00000E+00 (1/Pa/K) **3

FATIGUE PARAMETERS

BEARING WEIBULL SLOPE: 1.11111E+00
LIFE MODIFICATION CODE: 1

	......LINE CONTACT....... FATIGUE CONS (N/m**(50/27))	LOAD EXP	......POINT CONTACT...... FATIGUE CONS (N/m**1.80)	LOAD EXP	WEIBULL DISPERSION EXPONENT
OUTER RACE	1.66550E+08	4.00000E+00	1.87431E+07	3.00000E+00	1.11111E+00
INNER RACE	1.66550E+08	4.00000E+00	1.87431E+07	3.00000E+00	1.11111E+00

	.........ASPERITY INTERACTIONS........ COMPOSITE RMS HEIGHT (m)	COMPOSITE RMS SLOPE (rad)	TRACTION COEFF	LIMIT SHEAR STRESS (Pa)	RESIDUAL STRESS (Pa)	MATERIALS FACTOR	CONTAM-INATION FACTOR	PROCESSING FACTOR
OUTER RACE	1.00000E-07	4.11320E-02	1.00000E-01	0.00000E+00	0.00000E+00	1.19700E+00	1.00000E+00	7.70000E-02
INNER RACE	1.00000E-07	4.11320E-02	1.00000E-01	0.00000E+00	0.00000E+00	1.19700E+00	1.00000E+00	7.70000E-02

INITIAL OPERATING CONDITIONS

Ball Bearing Test Case kGeoMod=100

ROOM TEMPERATURE	(K)	3.23000E+02
HOUSING TEMPERATURE	(K)	3.23000E+02
SHAFT TEMPERATURE	(K)	3.23000E+02
ROLLING ELEMENT TEMPERATURE	(K)	3.23000E+02
CAGE TEMPERATURE	(K)	3.23000E+02

		OUTER RACE	INNER RACE
ANGULAR VELOCITY	(rpm)	0.00000E+00	2.00000E+04
TEMPERATURE	(K)	3.23000E+02	3.23000E+02
MISALIGNMENT-Y	(rad)	0.00000E+00	0.00000E+00
MISALIGNMENT-Z	(rad)	0.00000E+00	0.00000E+00

```
QUASI-STATIC CONSTRAINTS            0 1 0 1 1                TRANSLATIONAL CONSTRAINTS        1   1   1    1   1   1
CONSTRAINING LOAD FRACTION          0.00000E+00              ROTATIONAL CONSTRAINTS           1   1   1    1   1   1

BEARING COOLANT TYPE:                 Lubricant              COOLANT FLOW RATE     (m**3/s)  1.00000E-05
VARY GEOMETRY WITH TEMPERATURE:             Yes              COOLANT INLET TEMPERATURE  (K)  3.23000E+02
HEAT TRANS COEFF         (W/m**2/K)     Compute


                                          X-COMP       Y-COMP       Z-COMP

APPLIED LOAD VECTOR               (N)  4.50000E+03  0.00000E+00  2.00000E+03
RELATIVE DISPLACEMENT VECTOR      (m)  0.00000E+00  0.00000E+00  0.00000E+00
APPLIED MOMENT VECTOR           (N.m)               0.00000E+00  0.00000E+00
GRAVITY VECTOR                (m/s**2) 0.00000E+00  0.00000E+00 -9.81000E+00
CAGE GEOM CEN POSITION            (m)  0.00000E+00  0.00000E+00 -6.00000E-04
CAGE ANGULAR POSITION           (deg)  0.00000E+00  0.00000E+00  0.00000E+00
CAGE WHIRL TO ANG VEL RATIO            1.00000E+00
CAGE ANG VEL TO EPICYCLIC VEL RATIO    1.00000E+00


SCALE FACTORS AND OUTPUT CONTROLS
---------------------------------

         ......SCALE FACTORS.......          ......STEP SIZE INFO......           NO OF STEPS   2500
         LENGTH    (m)  9.52500E-03          MINIMUM        5.00000E-03           DATA CONTROL    1    500
         LOAD      (N)  4.50000E+03          MAXIMUM        3.00000E-01           AUTO PLOTS      1     19      0
         TIME      (S)  2.43679E-04          INITIAL        5.00000E-02                           0      0      0
                                             ERROR LIMIT    1.00000E-04           INT METHOD      5

step        tau          Time        Outer Race Rot  Inner Race Rot     (ADORE 5.0)
 no                       (s)            (rev)           (rev)
  0      0.000E+00     0.000E+00      0.000E+00       0.000E+00       Ball Bearing Test Case kGeoMod=100
=====    ==========    ==========     ==========      ==========      ====================================
```

1. Rolling Element Parameters

Re no	...Orbital Position (deg)	....Contact Angle.... (deg) Outer Race	Inner Race	....Contact Load..... (N) Outer Race	Inner Race	...Contact Stress.... (Pa) Outer Race	Inner Race	..Major Half Width... (m) Outer Race	Inner Race	..Minor Half Width... (m) Outer Race	Inner Race
1	0.000E+00	8.472E+00	2.745E+01	2.474E+03	7.907E+02	1.581E+09	1.399E+09	2.372E-03	1.260E-03	3.149E-04	2.142E-04
2	2.000E+01	8.334E+00	2.753E+01	2.457E+03	7.704E+02	1.577E+09	1.386E+09	2.367E-03	1.249E-03	3.142E-04	2.123E-04
3	4.000E+01	7.936E+00	2.778E+01	2.411E+03	7.142E+02	1.568E+09	1.352E+09	2.352E-03	1.218E-03	3.122E-04	2.071E-04
4	6.000E+01	7.327E+00	2.816E+01	2.348E+03	6.345E+02	1.554E+09	1.299E+09	2.331E-03	1.171E-03	3.095E-04	1.991E-04
5	8.000E+01	6.591E+00	2.862E+01	2.279E+03	5.460E+02	1.538E+09	1.235E+09	2.308E-03	1.114E-03	3.065E-04	1.895E-04
6	1.000E+02	5.816E+00	2.911E+01	2.218E+03	4.621E+02	1.524E+09	1.168E+09	2.288E-03	1.054E-03	3.037E-04	1.793E-04
7	1.200E+02	5.101E+00	2.956E+01	2.170E+03	3.912E+02	1.513E+09	1.105E+09	2.271E-03	9.966E-04	3.016E-04	1.696E-04
8	1.400E+02	4.505E+00	2.993E+01	2.153E+03	3.389E+02	1.509E+09	1.053E+09	2.265E-03	9.500E-04	3.008E-04	1.618E-04
9	1.600E+02	4.139E+00	3.016E+01	2.133E+03	3.064E+02	1.504E+09	1.018E+09	2.258E-03	9.186E-04	2.999E-04	1.565E-04
10	1.800E+02	4.014E+00	3.024E+01	2.127E+03	2.955E+02	1.503E+09	1.006E+09	2.256E-03	9.076E-04	2.996E-04	1.546E-04
11	2.000E+02	4.139E+00	3.016E+01	2.133E+03	3.064E+02	1.504E+09	1.018E+09	2.258E-03	9.186E-04	2.999E-04	1.565E-04
12	2.200E+02	4.505E+00	2.993E+01	2.153E+03	3.389E+02	1.509E+09	1.053E+09	2.265E-03	9.500E-04	3.008E-04	1.618E-04
13	2.400E+02	5.101E+00	2.956E+01	2.170E+03	3.912E+02	1.513E+09	1.105E+09	2.271E-03	9.966E-04	3.016E-04	1.696E-04
14	2.600E+02	5.816E+00	2.911E+01	2.218E+03	4.621E+02	1.524E+09	1.168E+09	2.288E-03	1.054E-03	3.037E-04	1.793E-04
15	2.800E+02	6.591E+00	2.862E+01	2.279E+03	5.460E+02	1.538E+09	1.235E+09	2.308E-03	1.114E-03	3.065E-04	1.895E-04
16	3.000E+02	7.327E+00	2.816E+01	2.348E+03	6.345E+02	1.554E+09	1.299E+09	2.331E-03	1.171E-03	3.095E-04	1.991E-04
17	3.200E+02	7.936E+00	2.778E+01	2.411E+03	7.142E+02	1.568E+09	1.352E+09	2.352E-03	1.218E-03	3.122E-04	2.071E-04
18	3.400E+02	8.334E+00	2.753E+01	2.457E+03	7.704E+02	1.577E+09	1.386E+09	2.367E-03	1.249E-03	3.142E-04	2.123E-04

RE	...Orbital Velocity (rpm)	........Angular Velocity........ Amplitude (rpm)	Theta (deg)	Phi (deg)	...RE Ang Position... Theta (deg)	Phi (deg)	......Spin/Roll...... Outer Race	Inner Race	....Contact Loss..... (W) Outer Race	Inner Race	..Time Ave Wear Rate (m**3/s)
1	8.999E+03	7.523E+04	-8.452E+00	0.000E+00	0.000E+00	0.000E+00	-1.958E-02	3.654E-01	1.848E+01	3.165E+01	1.683E-12
2	9.006E+03	7.529E+04	-8.313E+00	0.000E+00	0.000E+00	0.000E+00	-1.926E-02	3.695E-01	1.803E+01	3.030E+01	1.652E-12
3	9.028E+03	7.547E+04	-7.916E+00	0.000E+00	0.000E+00	0.000E+00	-1.835E-02	3.812E-01	1.686E+01	2.667E+01	1.566E-12
4	9.061E+03	7.576E+04	-7.309E+00	0.000E+00	0.000E+00	0.000E+00	-1.695E-02	3.994E-01	1.535E+01	2.178E+01	1.443E-12
5	9.100E+03	7.617E+04	-6.572E+00	0.000E+00	0.000E+00	0.000E+00	-1.517E-02	4.218E-01	1.381E+01	1.674E+01	1.286E-12
6	9.147E+03	7.659E+04	-5.798E+00	0.000E+00	0.000E+00	0.000E+00	-1.339E-02	4.458E-01	1.260E+01	1.238E+01	1.152E-12
7	9.194E+03	7.700E+04	-5.085E+00	0.000E+00	0.000E+00	0.000E+00	-1.174E-02	4.684E-01	1.174E+01	9.077E+00	1.040E-12
8	9.272E+03	7.765E+04	-3.612E+00	0.000E+00	0.000E+00	0.000E+00	7.041E-03	5.042E-01	1.101E+01	7.225E+00	9.767E-13
9	9.297E+03	7.785E+04	-3.319E+00	0.000E+00	0.000E+00	0.000E+00	6.469E-03	5.150E-01	1.064E+01	5.920E+00	9.234E-13
10	9.306E+03	7.792E+04	-3.218E+00	0.000E+00	0.000E+00	0.000E+00	6.271E-03	5.187E-01	1.052E+01	5.507E+00	9.056E-13
11	9.297E+03	7.785E+04	-3.319E+00	0.000E+00	0.000E+00	0.000E+00	6.469E-03	5.150E-01	1.064E+01	5.920E+00	9.234E-13
12	9.272E+03	7.765E+04	-3.612E+00	0.000E+00	0.000E+00	0.000E+00	7.041E-03	5.042E-01	1.101E+01	7.225E+00	9.767E-13
13	9.194E+03	7.700E+04	-5.085E+00	0.000E+00	0.000E+00	0.000E+00	-1.174E-02	4.684E-01	1.174E+01	9.077E+00	1.040E-12
14	9.147E+03	7.659E+04	-5.798E+00	0.000E+00	0.000E+00	0.000E+00	-1.339E-02	4.458E-01	1.260E+01	1.238E+01	1.152E-12
15	9.100E+03	7.617E+04	-6.572E+00	0.000E+00	0.000E+00	0.000E+00	-1.517E-02	4.218E-01	1.381E+01	1.674E+01	1.286E-12
16	9.061E+03	7.576E+04	-7.309E+00	0.000E+00	0.000E+00	0.000E+00	-1.695E-02	3.994E-01	1.535E+01	2.178E+01	1.443E-12
17	9.028E+03	7.547E+04	-7.916E+00	0.000E+00	0.000E+00	0.000E+00	-1.835E-02	3.812E-01	1.686E+01	2.667E+01	1.566E-12
18	9.006E+03	7.529E+04	-8.313E+00	0.000E+00	0.000E+00	0.000E+00	-1.926E-02	3.695E-01	1.803E+01	3.030E+01	1.652E-12

RE no	....Slip Velocity.... (m/s) Outer Race	Inner Race	.....Trac Coeff...... Outer Race	Inner Race	....Iso Lub Film..... (m) Outer Race	Inner Race	...Thermal Red Fac... Outer Race	Inner Race	......Drag (N)	..Chrn Mom (N.m)	..Net Loss (W) (Drag+Chur)

1	1.978E+00	4.216E+00	7.673E-03	1.267E-02	1.293E-06	1.188E-06	3.559E-01	3.908E-01
2	1.967E+00	4.212E+00	7.608E-03	1.253E-02	1.294E-06	1.190E-06	3.559E-01	3.919E-01
3	1.937E+00	4.197E+00	7.433E-03	1.214E-02	1.298E-06	1.194E-06	3.559E-01	3.951E-01
4	1.895E+00	4.169E+00	7.191E-03	1.156E-02	1.304E-06	1.202E-06	3.557E-01	4.000E-01
5	1.785E+00	4.122E+00	6.412E-03	1.087E-02	1.311E-06	1.212E-06	3.543E-01	4.057E-01
6	1.730E+00	4.055E+00	6.096E-03	1.016E-02	1.318E-06	1.223E-06	3.535E-01	4.121E-01
7	1.683E+00	3.973E+00	5.834E-03	9.503E-03	1.325E-06	1.234E-06	3.524E-01	4.182E-01
8	1.637E+00	4.014E+00	5.672E-03	9.171E-03	1.333E-06	1.241E-06	3.502E-01	4.244E-01
9	1.626E+00	3.937E+00	5.611E-03	8.796E-03	1.336E-06	1.248E-06	3.497E-01	4.277E-01
10	1.622E+00	3.909E+00	5.591E-03	8.665E-03	1.337E-06	1.250E-06	3.495E-01	4.289E-01
11	1.626E+00	3.937E+00	5.611E-03	8.796E-03	1.336E-06	1.248E-06	3.497E-01	4.277E-01
12	1.637E+00	4.014E+00	5.672E-03	9.171E-03	1.333E-06	1.241E-06	3.502E-01	4.244E-01
13	1.683E+00	3.973E+00	5.834E-03	9.503E-03	1.325E-06	1.234E-06	3.524E-01	4.182E-01
14	1.730E+00	4.055E+00	6.096E-03	1.016E-02	1.318E-06	1.223E-06	3.535E-01	4.121E-01
15	1.785E+00	4.122E+00	6.412E-03	1.087E-02	1.311E-06	1.212E-06	3.543E-01	4.057E-01
16	1.895E+00	4.169E+00	7.191E-03	1.156E-02	1.304E-06	1.202E-06	3.557E-01	4.000E-01
17	1.937E+00	4.197E+00	7.433E-03	1.214E-02	1.298E-06	1.194E-06	3.559E-01	3.951E-01
18	1.967E+00	4.212E+00	7.608E-03	1.253E-02	1.294E-06	1.190E-06	3.559E-01	3.919E-01

RE no	Rel Axial Pos (m)	.Contact Deflection.. (m) Outer Race	Inner Race	....Race Flexing..... (m) Outer Race	Inner Race	..Contact Temp Rise.. (K) Outer Race	Inner Race	.....Contact Temp.... (K) Outer Race	Inner Race	Bulk Temp (K)	Convec HTC (W/m**2/K)
1	-7.498E-05	1.595E-05	8.916E-06	0.000E+00	0.000E+00	6.288E-01	2.437E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
2	-7.594E-05	1.588E-05	8.762E-06	0.000E+00	0.000E+00	6.159E-01	2.365E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
3	-7.870E-05	1.568E-05	8.330E-06	0.000E+00	0.000E+00	5.816E-01	2.167E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
4	-8.291E-05	1.540E-05	7.698E-06	0.000E+00	0.000E+00	5.369E-01	1.883E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
5	-8.800E-05	1.510E-05	6.964E-06	0.000E+00	0.000E+00	4.807E-01	1.567E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
6	-9.336E-05	1.483E-05	6.230E-06	0.000E+00	0.000E+00	4.424E-01	1.266E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
7	-9.829E-05	1.462E-05	5.575E-06	0.000E+00	0.000E+00	4.148E-01	1.014E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
8	-1.024E-04	1.454E-05	5.067E-06	0.000E+00	0.000E+00	3.928E-01	8.725E-01	3.230E+02	3.230E+02	3.230E+02	0.000E+00
9	-1.049E-04	1.445E-05	4.737E-06	0.000E+00	0.000E+00	3.814E-01	7.542E-01	3.230E+02	3.230E+02	3.230E+02	0.000E+00
10	-1.058E-04	1.442E-05	4.624E-06	0.000E+00	0.000E+00	3.778E-01	7.152E-01	3.230E+02	3.230E+02	3.230E+02	0.000E+00
11	-1.049E-04	1.445E-05	4.737E-06	0.000E+00	0.000E+00	3.814E-01	7.542E-01	3.230E+02	3.230E+02	3.230E+02	0.000E+00
12	-1.024E-04	1.454E-05	5.067E-06	0.000E+00	0.000E+00	3.928E-01	8.725E-01	3.230E+02	3.230E+02	3.230E+02	0.000E+00
13	-9.829E-05	1.462E-05	5.575E-06	0.000E+00	0.000E+00	4.148E-01	1.014E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
14	-9.336E-05	1.483E-05	6.230E-06	0.000E+00	0.000E+00	4.424E-01	1.266E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
15	-8.800E-05	1.510E-05	6.964E-06	0.000E+00	0.000E+00	4.807E-01	1.567E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
16	-8.291E-05	1.540E-05	7.698E-06	0.000E+00	0.000E+00	5.369E-01	1.883E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
17	-7.870E-05	1.568E-05	8.330E-06	0.000E+00	0.000E+00	5.816E-01	2.167E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00
18	-7.594E-05	1.588E-05	8.762E-06	0.000E+00	0.000E+00	6.159E-01	2.365E+00	3.230E+02	3.230E+02	3.230E+02	0.000E+00

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Ball Bearing Test Case kGeoMod=100

2. Race and Cage Parameters

RE NO	C C	....RE/Cage Interaction.... Geo Int (m)	Con Ang Force (N)	Con Ang THETA (deg)	Con Ang PHI (deg)	Contact Pos X (m)	Contact Pos Z (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	0	5.844E-04	0.000E+00	2.700E+02	0.000E+00	9.541E-03	1.544E-03	0.000E+00	0.000E+00
2	0	3.977E-04	0.000E+00	3.558E+02	0.000E+00	7.055E-04	1.508E-03	0.000E+00	0.000E+00
3	0	2.206E-04	0.000E+00	3.582E+02	0.000E+00	3.110E-04	1.406E-03	0.000E+00	0.000E+00
4	0	8.900E-05	0.000E+00	3.591E+02	0.000E+00	1.513E-04	1.249E-03	0.000E+00	0.000E+00
5	0	1.898E-05	0.000E+00	3.597E+02	0.000E+00	4.552E-05	1.057E-03	0.000E+00	0.000E+00
6	0	1.897E-05	0.000E+00	2.702E-01	0.000E+00	-4.766E-05	8.522E-04	0.000E+00	0.000E+00
7	0	8.900E-05	0.000E+00	8.609E-01	0.000E+00	-1.508E-04	6.600E-04	0.000E+00	0.000E+00
8	0	2.206E-04	0.000E+00	1.780E+00	0.000E+00	-3.076E-04	5.033E-04	0.000E+00	0.000E+00
9	0	3.977E-04	0.000E+00	4.055E+00	0.000E+00	-6.879E-04	4.009E-04	0.000E+00	0.000E+00
10	0	5.848E-04	0.000E+00	9.000E+01	0.000E+00	-9.540E-03	3.654E-04	0.000E+00	0.000E+00
11	0	3.977E-04	0.000E+00	1.759E+02	0.000E+00	-6.879E-04	4.009E-04	0.000E+00	0.000E+00
12	0	2.206E-04	0.000E+00	1.782E+02	0.000E+00	-3.076E-04	5.033E-04	0.000E+00	0.000E+00
13	0	8.900E-05	0.000E+00	1.791E+02	0.000E+00	-1.508E-04	6.600E-04	0.000E+00	0.000E+00
14	0	1.897E-05	0.000E+00	1.797E+02	0.000E+00	-4.766E-05	8.522E-04	0.000E+00	0.000E+00
15	0	1.898E-05	0.000E+00	1.803E+02	0.000E+00	4.552E-05	1.057E-03	0.000E+00	0.000E+00
16	0	8.900E-05	0.000E+00	1.809E+02	0.000E+00	1.513E-04	1.249E-03	0.000E+00	0.000E+00
17	0	2.206E-04	0.000E+00	1.818E+02	0.000E+00	3.110E-04	1.406E-03	0.000E+00	0.000E+00
18	0	3.977E-04	0.000E+00	1.842E+02	0.000E+00	7.055E-04	1.508E-03	0.000E+00	0.000E+00

LD NO	C C	....Race/Cage Interaction.... Geo Int (m)	Force (N)	Con Angle (deg)	Att Angle (deg)	Effective Dia Play (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	0	1.217E-04	0.000E+00	0.000E+00	0.000E+00	1.443E-03	0.000E+00	0.000E+00
2	0	1.217E-04	0.000E+00	0.000E+00	0.000E+00	1.443E-03	0.000E+00	0.000E+00

	......Mass Center Position..... Axial (m)	Radial (m)	Orbital (deg)	...Orbital Velocity (rpm)	........Angular Velocity........ Amplitude (rpm)	Theta (deg)	Phi (deg)	....Ang Position..... Theta (deg)	Phi (deg)	......Hoop Stress (Pa)	..Time Ave Wear Rate (m**3/s)
Cage	4.285E-05	5.900E-04	1.800E+02	9.140E+03	9.140E+03	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-6.402E+06	1.110E-11
IRace	4.138E-04	1.002E-05	0.000E+00	0.000E+00	2.000E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.184E+08	1.157E-11

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Ball Bearing Test Case kGeoMod=100

3. Applied Parameters

	....Forces in Inertial Frame.... Comp-x (N)	Comp-y (N)	Comp-z (N)	...Moments in Body-Fixed Frame... Comp-x (N.m)	Comp-y (N.m)	Comp-z (N.m)
Cage	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
Int ORace	4.495E+03	-7.394E+01	1.453E+03	-3.472E-01	6.726E+01	4.428E-03
Int IRace	-4.495E+03	-4.926E-01	-1.995E+03	-1.004E-01	-6.827E+01	-3.967E-02
Ext ORace	-4.495E+03	7.394E+01	-1.440E+03	3.472E-01	-6.726E+01	-4.428E-03
Ext IRace	4.495E+03	4.926E-01	2.003E+03	1.004E-01	6.827E+01	3.967E-02

Basic Fatigue Life	(hours)	5.334E+02
Modified Fatigue Life	(hours)	4.460E+03
Outer Race Fit	(m)	1.000E-05
Inner Race Fit	(m)	9.918E-07
Internal Clearance	(m)	1.525E-04
Total Power Loss	(W)	5.464E+02
Churning Loss Fraction		0.000E+00
Exit Fluid Temperature	(K)	0.000E+00

	....MC Acc in Inertial Frame.... (m/s**2) Comp-x	Comp-y	Comp-z	...Ang Acc in Body-Fixed Frame... (rpm/s) Comp-x	Comp-y	Comp-z	Bulk Temp (K)
Cage	0.000E+00	0.000E+00	-9.810E+00	0.000E+00	0.000E+00	0.000E+00	3.230E+02
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.230E+02
IRace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.230E+02

Hsng/Bulk Temp	(K)	3.230E+02
Hsng/Outer Temp	(K)	3.230E+02
Hsng/Inner Temp	(K)	3.230E+02
Shft/Bulk Temp	(K)	3.230E+02
Shft/Outer Temp	(K)	3.230E+02
Shft/Inner Temp	(K)	3.230E+02

4. Time Step Summary

Step no	Time (s)	Outer Race Rotation (rev)	Inner Race Rotation (rev)	...Time Average Parameters... Fatigue Life (hours)	Power Loss (W)	RE Orbital Vel Ratio	Cage Omega Ratio	Cage Whirl Ratio	Cage Wear Rate (m**3/s)	Average RE Excursion (m)
0	0.000E+00	0.000E+00	0.000E+00	4.460E+03	5.464E+02	4.570E-01	4.570E-01	4.570E-01	0.000E+00	1.174E-03

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
2500	3.534E+02	8.612E-02	0.000E+00	2.871E+01	Ball Bearing Test Case kGeoMod=100

1. Rolling Element Parameters

Re no	...Orbital Position (deg)	....Contact Angle.... (deg) Outer Race	Inner Race	....Contact Load..... (N) Outer Race	Inner Race	...Contact Stress.... (Pa) Outer Race	Inner Race	..Major Half Width... (m) Outer Race	Inner Race	..Minor Half Width... (m) Outer Race	Inner Race
1	4.347E+03	7.675E+00	2.782E+01	2.416E+03	6.914E+02	1.568E+09	1.337E+09	2.354E-03	1.205E-03	3.125E-04	2.049E-04
2	4.367E+03	7.241E+00	2.809E+01	2.359E+03	6.316E+02	1.556E+09	1.297E+09	2.335E-03	1.169E-03	3.100E-04	1.988E-04
3	4.387E+03	6.708E+00	2.843E+01	2.293E+03	5.626E+02	1.541E+09	1.248E+09	2.313E-03	1.125E-03	3.071E-04	1.914E-04
4	4.407E+03	6.127E+00	2.881E+01	2.229E+03	4.937E+02	1.527E+09	1.194E+09	2.291E-03	1.077E-03	3.042E-04	1.833E-04
5	4.427E+03	5.592E+00	2.916E+01	2.172E+03	4.344E+02	1.513E+09	1.144E+09	2.271E-03	1.032E-03	3.016E-04	1.756E-04
6	4.447E+03	5.120E+00	2.947E+01	2.137E+03	3.876E+02	1.505E+09	1.101E+09	2.259E-03	9.936E-04	3.000E-04	1.691E-04
7	4.467E+03	4.830E+00	2.967E+01	2.113E+03	3.594E+02	1.499E+09	1.074E+09	2.251E-03	9.688E-04	2.989E-04	1.650E-04
8	4.487E+03	4.596E+00	2.982E+01	2.084E+03	3.358E+02	1.493E+09	1.050E+09	2.241E-03	9.471E-04	2.976E-04	1.613E-04
9	4.507E+03	4.854E+00	2.969E+01	2.108E+03	3.601E+02	1.498E+09	1.074E+09	2.249E-03	9.694E-04	2.987E-04	1.651E-04
10	4.527E+03	4.743E+00	2.973E+01	2.104E+03	3.508E+02	1.497E+09	1.065E+09	2.248E-03	9.610E-04	2.985E-04	1.636E-04
11	4.547E+03	5.091E+00	2.950E+01	2.139E+03	3.855E+02	1.506E+09	1.099E+09	2.260E-03	9.917E-04	3.002E-04	1.688E-04
12	4.567E+03	5.555E+00	2.920E+01	2.185E+03	4.336E+02	1.516E+09	1.143E+09	2.276E-03	1.031E-03	3.023E-04	1.755E-04
13	4.588E+03	6.114E+00	2.883E+01	2.245E+03	4.957E+02	1.530E+09	1.196E+09	2.297E-03	1.079E-03	3.050E-04	1.835E-04
14	4.608E+03	6.703E+00	2.845E+01	2.311E+03	5.661E+02	1.545E+09	1.250E+09	2.319E-03	1.127E-03	3.079E-04	1.918E-04
15	4.628E+03	7.264E+00	2.809E+01	2.360E+03	6.336E+02	1.556E+09	1.298E+09	2.335E-03	1.171E-03	3.100E-04	1.990E-04
16	4.648E+03	7.639E+00	2.785E+01	2.415E+03	6.871E+02	1.568E+09	1.334E+09	2.353E-03	1.203E-03	3.124E-04	2.045E-04
17	4.668E+03	7.914E+00	2.768E+01	2.454E+03	7.276E+02	1.577E+09	1.360E+09	2.366E-03	1.226E-03	3.141E-04	2.084E-04
18	4.688E+03	8.352E+00	2.744E+01	2.506E+03	7.900E+02	1.588E+09	1.398E+09	2.383E-03	1.260E-03	3.163E-04	2.141E-04

RE	...Orbital Velocity (rpm)	........Angular Velocity........ Amplitude (rpm)	Theta (deg)	Phi (deg)	...RE Ang Position... Theta (deg)	Phi (deg)	......Spin/Roll...... Outer Race	Inner Race	....Contact Loss..... (W) Outer Race	Inner Race	..Time Ave Wear Rate (m**3/s)
1	9.094E+03	7.611E+04	-6.154E+00	6.906E-05	1.012E+01	1.447E+02	1.201E-02	4.129E-01	1.632E+01	2.735E+01	2.149E-13
2	9.096E+03	7.612E+04	-6.074E+00	1.666E-05	1.402E+01	1.936E+02	6.020E-03	4.199E-01	1.457E+01	2.271E+01	2.054E-13
3	9.094E+03	7.611E+04	-6.141E+00	3.600E+02	1.374E+01	1.931E+02	-4.609E-03	4.257E-01	1.307E+01	1.822E+01	2.045E-13
4	9.096E+03	7.612E+04	-6.186E+00	3.600E+02	1.325E+01	1.929E+02	-1.563E-02	4.326E-01	1.255E+01	1.432E+01	2.037E-13
5	9.092E+03	7.610E+04	-6.335E+00	3.600E+02	1.249E+01	1.942E+02	-2.794E-02	4.372E-01	1.316E+01	1.145E+01	2.031E-13
6	9.115E+03	7.629E+04	-6.340E+00	3.600E+02	1.181E+01	1.928E+02	-3.627E-02	4.436E-01	1.413E+01	8.797E+00	2.038E-13
7	9.121E+03	7.635E+04	-6.346E+00	3.600E+02	1.116E+01	1.912E+02	-4.146E-02	4.477E-01	1.480E+01	7.595E+00	2.030E-13
8	9.104E+03	7.622E+04	-6.331E+00	3.600E+02	9.697E+00	1.884E+02	-4.526E-02	4.512E-01	1.505E+01	7.474E+00	2.031E-13
9	9.107E+03	7.624E+04	-6.329E+00	3.600E+02	9.558E+00	1.863E+02	-4.072E-02	4.485E-01	1.450E+01	8.171E+00	2.013E-13
10	9.118E+03	7.633E+04	-6.202E+00	3.600E+02	9.284E+00	1.866E+02	-4.012E-02	4.516E-01	1.431E+01	7.608E+00	2.014E-13
11	9.127E+03	7.639E+04	-6.095E+00	3.600E+02	9.338E+00	1.849E+02	-3.192E-02	4.487E-01	1.334E+01	8.763E+00	1.999E-13

12	9.130E+03	7.641E+04	-6.084E+00	3.600E+02	9.519E+00	1.872E+02	-2.361E-02	4.426E-01	1.282E+01	1.079E+01	2.012E-13
13	9.132E+03	7.642E+04	-6.010E+00	3.600E+02	1.092E+01	1.879E+02	-1.239E-02	4.363E-01	1.264E+01	1.411E+01	1.993E-13
14	9.131E+03	7.641E+04	-5.998E+00	3.600E+02	1.164E+01	1.881E+02	-1.867E-03	4.286E-01	1.346E+01	1.880E+01	1.990E-13
15	9.092E+03	7.610E+04	-5.972E+00	1.996E-03	1.234E+01	1.895E+02	8.449E-03	4.218E-01	1.477E+01	2.305E+01	2.009E-13
16	9.102E+03	7.617E+04	-5.953E+00	3.600E+02	1.299E+01	1.930E+02	1.535E-02	4.170E-01	1.669E+01	2.725E+01	2.091E-13
17	9.104E+03	7.619E+04	-6.113E+00	3.600E+02	1.379E+01	1.924E+02	1.700E-02	4.107E-01	1.791E+01	3.093E+01	2.042E-13
18	9.084E+03	7.603E+04	-6.093E+00	3.600E+02	1.415E+01	1.918E+02	2.505E-02	4.064E-01	2.066E+01	3.553E+01	2.044E-13

RE no	....Slip Velocity.... (m/s) Outer Race	Inner Race	.....Trac Coeff...... Outer Race	Inner Race	....Iso Lub Film..... (m) Outer Race	Inner Race	...Thermal Red Fac... Outer Race	Inner Race	......Drag (N)	..Chrn Mom (N.m)	..Net Loss (W) (Drag+Chur)
1	1.835E+00	4.277E+00	6.869E-03	1.203E-02	1.305E-06	1.194E-06	3.527E-01	3.949E-01			
2	1.717E+00	4.346E+00	6.274E-03	1.181E-02	1.307E-06	1.200E-06	3.535E-01	4.011E-01			
3	1.662E+00	4.379E+00	5.963E-03	1.163E-02	1.310E-06	1.209E-06	3.546E-01	4.019E-01			
4	1.788E+00	4.396E+00	6.451E-03	1.136E-02	1.313E-06	1.219E-06	3.557E-01	4.038E-01			
5	1.936E+00	4.402E+00	7.005E-03	1.109E-02	1.315E-06	1.230E-06	3.569E-01	4.055E-01			
6	2.045E+00	4.166E+00	7.403E-03	1.022E-02	1.318E-06	1.239E-06	3.567E-01	4.106E-01			
7	2.108E+00	4.120E+00	7.613E-03	9.941E-03	1.320E-06	1.245E-06	3.570E-01	4.124E-01			
8	2.140E+00	4.351E+00	7.684E-03	1.039E-02	1.320E-06	1.250E-06	3.581E-01	4.104E-01			
9	2.088E+00	4.314E+00	7.516E-03	1.044E-02	1.319E-06	1.245E-06	3.575E-01	4.099E-01			
10	2.079E+00	4.241E+00	7.473E-03	1.018E-02	1.320E-06	1.246E-06	3.572E-01	4.115E-01			
11	1.981E+00	4.175E+00	7.140E-03	1.019E-02	1.319E-06	1.239E-06	3.561E-01	4.113E-01			
12	1.887E+00	4.108E+00	6.815E-03	1.025E-02	1.318E-06	1.229E-06	3.552E-01	4.106E-01			
13	1.757E+00	4.065E+00	6.335E-03	1.042E-02	1.315E-06	1.219E-06	3.541E-01	4.089E-01			
14	1.639E+00	4.031E+00	5.885E-03	1.078E-02	1.313E-06	1.208E-06	3.530E-01	4.013E-01			
15	1.742E+00	4.449E+00	6.352E-03	1.210E-02	1.307E-06	1.200E-06	3.534E-01	3.994E-01			
16	1.886E+00	4.307E+00	7.095E-03	1.205E-02	1.305E-06	1.194E-06	3.524E-01	3.954E-01			
17	1.934E+00	4.158E+00	7.363E-03	1.197E-02	1.304E-06	1.190E-06	3.517E-01	3.903E-01			
18	2.079E+00	4.393E+00	8.062E-03	1.280E-02	1.300E-06	1.183E-06	3.517E-01	3.892E-01			

RE no	Rel Axial Pos (m)	.Contact Deflection.. (m) Outer Race	Inner Race	....Race Flexing..... (m) Outer Race	Inner Race	..Contact Temp Rise.. (K) Outer Race	Inner Race	.....Contact Temp.... (K) Outer Race	Inner Race	Bulk Temp (K)	Convec HTC (W/m**2/K)
1	-8.067E-05	1.570E-05	8.152E-06	0.000E+00	0.000E+00	5.579E-01	2.265E+00	3.428E+02	3.433E+02	3.354E+02	2.620E+06
2	-8.367E-05	1.545E-05	7.674E-06	0.000E+00	0.000E+00	5.048E-01	1.972E+00	3.418E+02	3.471E+02	3.354E+02	2.620E+06
3	-8.737E-05	1.516E-05	7.105E-06	0.000E+00	0.000E+00	4.596E-01	1.678E+00	3.427E+02	3.535E+02	3.364E+02	2.618E+06
4	-9.139E-05	1.488E-05	6.512E-06	0.000E+00	0.000E+00	4.460E-01	1.409E+00	3.451E+02	3.606E+02	3.382E+02	2.617E+06
5	-9.511E-05	1.462E-05	5.978E-06	0.000E+00	0.000E+00	4.705E-01	1.199E+00	3.480E+02	3.663E+02	3.407E+02	2.614E+06
6	-9.838E-05	1.447E-05	5.541E-06	0.000E+00	0.000E+00	5.062E-01	9.800E-01	3.512E+02	3.707E+02	3.437E+02	2.612E+06
7	-1.004E-04	1.436E-05	5.268E-06	0.000E+00	0.000E+00	5.312E-01	8.794E-01	3.543E+02	3.735E+02	3.472E+02	2.610E+06
8	-1.020E-04	1.423E-05	5.036E-06	0.000E+00	0.000E+00	5.428E-01	8.876E-01	3.539E+02	3.680E+02	3.477E+02	2.611E+06
9	-1.003E-04	1.434E-05	5.275E-06	0.000E+00	0.000E+00	5.213E-01	9.399E-01	3.593E+02	3.691E+02	3.535E+02	2.611E+06
10	-1.010E-04	1.432E-05	5.184E-06	0.000E+00	0.000E+00	5.147E-01	8.889E-01	3.541E+02	3.582E+02	3.476E+02	2.613E+06
11	-9.861E-05	1.448E-05	5.521E-06	0.000E+00	0.000E+00	4.781E-01	9.801E-01	3.542E+02	3.530E+02	3.464E+02	2.614E+06
12	-9.540E-05	1.468E-05	5.971E-06	0.000E+00	0.000E+00	4.572E-01	1.139E+00	3.541E+02	3.483E+02	3.445E+02	2.615E+06
13	-9.152E-05	1.495E-05	6.529E-06	0.000E+00	0.000E+00	4.478E-01	1.390E+00	3.529E+02	3.438E+02	3.420E+02	2.616E+06
14	-8.744E-05	1.524E-05	7.134E-06	0.000E+00	0.000E+00	4.713E-01	1.726E+00	3.518E+02	3.407E+02	3.400E+02	2.617E+06
15	-8.356E-05	1.546E-05	7.691E-06	0.000E+00	0.000E+00	5.095E-01	1.998E+00	3.501E+02	3.386E+02	3.382E+02	2.618E+06
16	-8.094E-05	1.570E-05	8.118E-06	0.000E+00	0.000E+00	5.696E-01	2.264E+00	3.475E+02	3.373E+02	3.365E+02	2.619E+06
17	-7.904E-05	1.587E-05	8.434E-06	0.000E+00	0.000E+00	6.060E-01	2.490E+00	3.460E+02	3.404E+02	3.367E+02	2.618E+06
18	-7.613E-05	1.609E-05	8.910E-06	0.000E+00	0.000E+00	6.886E-01	2.745E+00	3.519E+02	3.489E+02	3.440E+02	2.616E+06

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
2500	3.534E+02	8.612E-02	0.000E+00	2.871E+01	Ball Bearing Test Case kGeoMod=100
=====	==========	==========	==========	==========	===================================

2. Race and Cage Parameters

RE NO C	C	...........RE/Cage Interaction........... Geo Int (m)	Con Ang Force (N)	Con Ang THETA (deg)	Con Ang PHI (deg)	Contact Pos X (m)	Contact Pos Z (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	0	5.830E-04	0.000E+00	3.463E+02	0.000E+00	2.261E-03	7.258E-04	0.000E+00	6.826E-12
2	0	3.686E-04	0.000E+00	1.785E+02	0.000E+00	-2.535E-04	7.000E-04	0.000E+00	2.838E-12
3	0	3.859E-04	0.000E+00	1.720E+02	0.000E+00	-1.360E-03	7.081E-04	0.000E+00	2.148E-12
4	0	4.971E-04	0.000E+00	1.423E+02	0.000E+00	-5.887E-03	7.481E-04	0.000E+00	2.878E-12
5	0	4.770E-06	0.000E+00	1.009E+01	0.000E+00	-1.774E-03	8.107E-04	0.000E+00	3.116E-12
6	0	8.030E-05	0.000E+00	1.640E+01	0.000E+00	-2.838E-03	8.977E-04	0.000E+00	3.386E-12
7	0	1.728E-04	0.000E+00	2.573E+01	0.000E+00	-4.323E-03	9.940E-04	0.000E+00	3.926E-12
8	0	5.334E-05	0.000E+00	2.337E+01	0.000E+00	-3.997E-03	1.086E-03	0.000E+00	5.003E-12
9	0	3.130E-04	0.000E+00	5.465E+01	0.000E+00	-8.007E-03	1.167E-03	0.000E+00	5.094E-12
10	0	3.101E-04	0.000E+00	5.597E+01	0.000E+00	-8.138E-03	1.224E-03	0.000E+00	5.149E-12
11	0	2.266E-04	0.000E+00	1.419E+02	0.000E+00	-6.107E-03	1.250E-03	0.000E+00	3.916E-12
12	0	2.107E-04	0.000E+00	3.212E+01	0.000E+00	-5.275E-03	1.242E-03	0.000E+00	2.846E-12
13	0	1.697E-04	0.000E+00	1.565E+02	0.000E+00	-3.969E-03	1.201E-03	0.000E+00	1.696E-12
14	0	1.038E-04	0.000E+00	1.648E+02	0.000E+00	-2.628E-03	1.132E-03	0.000E+00	3.033E-12
15	0	1.181E-05	0.000E+00	1.715E+02	0.000E+00	-1.495E-03	1.044E-03	0.000E+00	2.686E-12
16	0	5.913E-05	0.000E+00	1.749E+02	0.000E+00	-8.948E-04	9.488E-04	0.000E+00	5.201E-12
17	0	2.713E-04	0.000E+00	1.770E+02	0.000E+00	-5.197E-04	8.581E-04	0.000E+00	4.808E-12
18	0	6.551E-05	0.000E+00	1.804E+02	0.000E+00	7.453E-05	7.765E-04	0.000E+00	5.625E-12

LD NO	C C	...........Race/Cage Interaction........... Geo Int	Force	Con Angle	Att Angle	Effective Dia Play	Contact Loss	Time Ave Wear Rate

```
                (m)        (N)      (deg)      (deg)        (m)        (W)   (m**3/s)

           1 0  4.446E-04  0.000E+00  2.043E+02  0.000E+00  1.472E-03  0.000E+00  8.772E-11
           2 0  4.446E-04  0.000E+00  2.043E+02  0.000E+00  1.472E-03  0.000E+00  8.772E-11


      ......Mass Center Position..... ...Orbital ........Angular Velocity........ ....Ang Position..... ......Hoop ..Time Ave
          Axial     Radial    Orbital   Velocity  Amplitude      Theta        Phi      Theta        Phi     Stress  Wear Rate
            (m)        (m)      (deg)      (rpm)      (rpm)      (deg)      (deg)      (deg)      (deg)       (Pa)   (m**3/s)

 Cage  1.608E-04  2.767E-04  5.259E+01  7.562E+03  9.108E+03  5.302E-02  1.868E+02  9.271E-02  1.785E+02  3.625E+07  2.456E-10
ORace  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00 -9.333E+06  1.792E-12
IRace  4.116E-04  9.355E-06  0.000E+00  0.000E+00  2.000E+04  0.000E+00  0.000E+00  0.000E+00  0.000E+00  1.167E+08  1.871E-12


  step         tau           Time       Outer Race Rot  Inner Race Rot     (ADORE 5.0)
   no                         (s)           (rev)           (rev)
  2500      3.534E+02      8.612E-02      0.000E+00       2.871E+01       Ball Bearing Test Case kGeoMod=100
 =====     ==========     ==========     ==========      ==========       ====================================


  3. Applied Parameters
  ---------------------
                                                                             Basic Fatigue Life    (hours)   5.515E+02
              ....Forces in Inertial Frame.... ...Moments in Body-Fixed Frame...  Modified Fatigue Life (hours)   4.594E+03
                   Comp-x     Comp-y     Comp-z     Comp-x     Comp-y     Comp-z  Outer Race Fit            (m)   1.458E-05
                      (N)        (N)        (N)      (N.m)      (N.m)      (N.m)  Inner Race Fit            (m)  -2.484E-05
                                                                             Internal Clearance        (m)   1.537E-04
        Cage     0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
    Int ORace    4.457E+03 -6.830E+01  1.694E+03 -1.153E-01  5.525E+01 -1.273E+00  Total Power Loss          (W)   5.677E+02
    Int IRace   -4.457E+03 -5.264E+01 -1.631E+03 -6.630E-01  1.426E+01 -5.424E+01  Churning Loss Fraction          0.000E+00
    Ext ORace   -4.457E+03  6.830E+01 -1.682E+03  1.153E-01 -5.525E+01  1.273E+00  Exit Fluid Temperature    (K)   3.419E+02
    Ext IRace    4.457E+03  5.264E+01  1.639E+03  6.630E-01 -1.426E+01  5.424E+01

              ....MC Acc in Inertial Frame.... ...Ang Acc in Body-Fixed Frame...  Bulk Temp  Hsng/Bulk Temp  (K)   3.468E+02
                              (m/s**2)                         (rpm/s)                  (K)  Hsng/Outer Temp (K)   3.457E+02
                   Comp-x     Comp-y     Comp-z     Comp-x     Comp-y     Comp-z             Hsng/Inner Temp (K)   3.480E+02
                                                                                        Shft/Bulk Temp  (K)   3.293E+02
        Cage     0.000E+00  0.000E+00 -9.810E+00  0.000E+00  7.266E+00  8.652E-01  3.420E+02
        ORace    0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  3.490E+02  Shft/Outer Temp (K)   3.496E+02
        IRace    0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  3.514E+02  Shft/Inner Temp (K)   3.174E+02


  4. Time Step Summary
  --------------------

                                            .....................Time Average Parameters.....................
 Step       Time Outer Race Inner Race    Fatigue      Power RE Orbital Cage Omega Cage Whirl       Cage Average RE
   no              Rotation   Rotation       Life       Loss  Vel Ratio      Ratio      Ratio  Wear Rate  Excursion
              (s)      (rev)      (rev)    (hours)        (W)                                   (m**3/s)        (m)

 2500  8.612E-02  0.000E+00  2.871E+01  6.952E+02  1.406E+02  7.024E-02  7.024E-02  7.306E-02  2.456E-10  1.841E-04

      Normal termination: Last Step Number =       2500     Final Time =   3.5340E+02

      Statistics of this Run
      ----------------------

      Minimum Step Size      =  5.00000E-03
      Maximum Step Size      =  1.86684E-01
      Last Step Size         =  1.64618E-01
      Max Truncation         =  1.72086E-03
      Total Derivative Calls =        21160
```

Appendix D - Cylindrical Roller Bearing Data

Adore input data for the roller bearing case with the associated print ouput is documented below. Only the output and initial and final time steps is included for brevity.

```
Listing of input data records
-----------------------------

Rec 1            1     0     0     0     0     1   500     2     0     0     1  4000     0     5
Rec 2.1      5.0000E-02 1.0000E-04 5.0000E-01 1.0000E+03 1.0000E-04 5.0000E-02
Rec 2.4          0     0     0     1
Rec 2.5      0.0000E+00 0.0000E+00 0.0000E+00 8.5700E-04 8.5700E-04
Rec 2.7      2.9400E+02 2.9400E+02 2.9400E+02 2.9400E+02 2.9400E+02 2.9400E+02
Rec 3.1     Cyl Roller Bearing Geo Mod
Rec 3.2       1  2  8  1  0  0  0  0  0  0  0  0
Rec 3.3       1  1  0  1  1  0  0  0  1  0  0  0  0  0  0
Rec 3.4       0  0  0  0  0  2  1  9  0  0  0  0
Rec 4        3.0000E-02 5.5000E-02 1.0000E-02 6.5000E-02 1.2000E-02 1.2000E-02
Rec 5B       8.0000E-03 2.5000E-01 8.0000E-03 2.0000E-03 2.5000E-04 2.5000E-04 4.3000E-02 4.0000E-05
Rec 5B.1     1.0000E-02 0.0000E+00 1.0000E-02 0.0000E+00
Rec 7.0       0  0  0  0  0  0  0  0  2  1  1
Rec 7.1      4.5000E-02 4.0000E-02 1.0000E-02 1.0000E-04 1.0000E-03 1.0000E-04 0.0000E+00 0.0000E+00 0.0000E+00
Rec 7.2.1    4.5000E-02 1.0000E-03 5.0000E-03 1.0000E-04
Rec 7.2.2    4.5000E-02 1.0000E-03 5.0000E-03 1.0000E-04
Rec 8.1      3.2000E+03 3.1000E+11 2.6000E-01 2.9000E-06 8.3700E+02 3.0500E+01 2.0000E-03 8.0000E+01 2.0000E-06
Rec 9.0      2.9400E+02 0.0000E+00 2.0000E-05
Rec 9.1.1    0.0000E+00 0.0000E+00 1.0000E+03 0.0000E+00 0.0000E+00 0.0000E+00
Rec 9.1.2    0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 7.0000E+04
Rec 9.5.1    2.5000E-04 1.0000E+00 1.0000E+00 0.0000E+00 0.0000E+00 5.0000E-05 0.0000E+00 0.0000E+00 0.0000E+00
Rec 10.0      0  1  0  0  0
Rec 10.1B    0.0000E+00 1.0000E-01 8.0000E-02 2.0000E-01
Rec 10.5.1B  5.0000E-02 0.0000E+00 0.0000E+00 0.0000E+00 1.2500E-07
Rec 10.5.2B  5.0000E-02 0.0000E+00 0.0000E+00 0.0000E+00 1.2500E-07
Rec 11       0.0000E+00 0.0000E+00-9.8100E+00

Input from optional user subroutines
------------------------------------
```

```
                                   ADVANCED DYNAMICS OF ROLLING ELEMENTS
                                   =====================================
                               A Real Time Simulation of Rolling Bearing Performance
                                          [ Version (ADORE 5.0) ]


                                                 Copyright
                                          Pradeep K Gupta Inc

*******************************************************************************************************************************
*                                                                                                                             *
*     Bearing Type = CYLINDRICAL ROLLER             Program Mode =  1       Run Id = Cyl Roller Bearing Geo Mod               *
*******************************************************************************************************************************


BEARING GEOMETRY
----------------

NO OF ROLLERS    8            PITCH DIA   (m) 4.30000E-02     OUTER FIT   (m) 0.00000E+00     SHAFT OD    (m) 3.00000E-02
ROLLER DIA  (m) 8.00000E-03   CROWN RAD   (m) 2.50000E-01     INNER FIT   (m) 2.00000E-05     SHAFT ID    (m) 1.00000E-02
ROLLER LEN  (m) 8.00000E-03   COR RAD  I  (m) 2.50000E-04     DIA PLAY    (m) 4.00000E-05     BEARING OD  (m) 5.50000E-02
CEN LAND    (m) 2.00000E-03   COR RAD II  (m) 2.50000E-04                                     HOUSING OD  (m) 6.50000E-02
OUTER LEN1  (m) 1.00000E-02   INNER LEN1  (m) 1.00000E-02     OUTER CRN   (m) 1.00000E+10     WIDTH  I    (m) 1.20000E-02
OUTER LEN2  (m) 0.00000E+00   INNER LEN2  (m) 0.00000E+00     INNER CRN   (m) 1.00000E+10     WIDTH II    (m) 1.20000E-02


CAGE OD     (m) 4.50000E-02   CAGE WIDTH  (m) 1.00000E-02     POC CLS  I  (m) 1.00000E-04     OUTER CLS   (m) 1.00000E-04
CAGE ID     (m) 4.00000E-02   SEG CUT   (deg) 0.00000E+00     POC CLS II  (m) 0.00000E+00     INNER CLS   (m) 1.00000E-03
CAGE CONE (deg) 0.00000E+00   POCKET SHAPE     0              NO OF C/R LANDS  2              NO OF CAGE SEGS  1

GUIDE  GUIDE     GUIDE LAND     GUIDE LAND     GUIDE LAND     GUIDE LAND
 LAND   TYPE            DIA            CLS          WIDTH       POSITION
                        (m)            (m)            (m)            (m)
```

1	1	4.50000E-02	1.00000E-04	1.00000E-03	5.00000E-03
2	1	4.50000E-02	1.00000E-04	1.00000E-03	5.00000E-03

MATERIAL PROPERTIES

		ROLLING ELEMENT	OUTER RACE	INNER RACE	CAGE	SHAFT	HOUSING
DENSITY	(kg/m**3)	3.20000E+03	7.75000E+03	7.75000E+03	7.75000E+03	7.75000E+03	7.75000E+03
ELASTIC MODULUS	(Pa)	3.10000E+11	2.00000E+11	2.00000E+11	2.00000E+11	2.00000E+11	2.00000E+11
POISSON-S RATIO		2.60000E-01	2.50000E-01	2.50000E-01	2.50000E-01	2.50000E-01	2.50000E-01
COEFF OF THERMAL EXP	(m/m/K)	2.90000E-06	1.17000E-05	1.17000E-05	1.80000E-05	1.17000E-05	1.17000E-05
THERMAL CONDUCTIVITY	(W/m/K)	3.05000E+01	4.30000E+01	4.30000E+01	4.30000E+01	4.30000E+01	4.30000E+01
HEAT CAPACITY	(J/kg/K)	8.37000E+02	4.70000E+02	4.70000E+02	4.70000E+02		
ELASTIC STRAIN LIMIT		2.00000E-03	2.00000E-03	2.00000E-03	2.00000E-03		
HARDNESS	(RC)	8.00000E+01	6.10000E+01	6.10000E+01	2.00000E+01		
WEAR COEFFICIENT		2.00000E-06	5.00000E-06	5.00000E-06	5.00000E-04		

INERTIAL PARAMETERS

	.......MASS (kg)	..........MOMENT OF INERTIA.......... (kg*m**2) X-COMP	Y-COMP	Z-COMP	.........MASS TO GEO CENTER.......... (m) X-COMP	Y-COMP	Z-COMP
RE	1.28108E-03	1.02486E-08	1.19567E-08	1.19567E-08	0.00000E+00	0.00000E+00	0.00000E+00
CAGE	1.58251E-02	7.17073E-06	3.71724E-06	3.71724E-06	0.00000E+00	0.00000E+00	0.00000E+00
OUTER RACE	3.08208E-02	2.16826E-05	1.12111E-05	1.12111E-05	0.00000E+00	0.00000E+00	0.00000E+00
INNER RACE	2.36364E-02	6.27429E-06	3.42078E-06	3.42078E-06	0.00000E+00	0.00000E+00	0.00000E+00

	.........PRINCIPAL TO GEO FRAME........ (deg) X-COMP	Y-COMP	Z-COMP
RE	0.00000E+00	0.00000E+00	0.00000E+00
CAGE	0.00000E+00	0.00000E+00	0.00000E+00
OUTER RACE	0.00000E+00	0.00000E+00	0.00000E+00
INNER RACE	0.00000E+00	0.00000E+00	0.00000E+00

LUBRICATION PARAMETERS

HYPOTHETICAL MODELS	CRITICAL FILM (m)	TRAC COEFF AT ZERO SLIP	MAXIMUM TRAC COEFF	TRAC COEFF AT INF SLIP	SLIP AT MAX TRACTION (m/s)	COEFFICIENT A	COEFFICIENT B (s/m)	COEFFICIENT C (s/m)	COEFFICIENT D
RE/RACE1	0.00000E+00	0.00000E+00	1.00000E-01	8.00000E-02	2.00000E-01	-8.00000E-02	9.57239E-01	8.58912E+00	8.00000E-02
RE/RACE2	0.00000E+00	0.00000E+00	1.00000E-01	8.00000E-02	2.00000E-01	-8.00000E-02	9.57239E-01	8.58912E+00	8.00000E-02
RE/CAGE	1.25000E-07				0.00000E+00	5.00000E-02	0.00000E+00	0.00000E+00	
RACE/CAGE	1.25000E-07				0.00000E+00	5.00000E-02	0.00000E+00	0.00000E+00	

FATIGUE PARAMETERS

	......LINE CONTACT....... FATIGUE CONS (N/m**(50/27))	LOAD EXP	......POINT CONTACT...... FATIGUE CONS (N/m**1.80)	LOAD EXP	WEIBULL DISPERSION EXPONENT
OUTER RACE	1.66550E+08	4.00000E+00	1.87431E+07	3.00000E+00	1.11111E+00
INNER RACE	1.66550E+08	4.00000E+00	1.87431E+07	3.00000E+00	1.11111E+00

BEARING WEIBULL SLOPE: 1.11111E+00
LIFE MODIFICATION CODE: 0

	.........ASPERITY INTERACTIONS........ COMPOSITE RMS HEIGHT (m)	COMPOSITE RMS SLOPE (rad)	TRACTION COEFF	LIMIT SHEAR STRESS (Pa)	RESIDUAL STRESS (Pa)	MATERIALS FACTOR	CONTAM-INATION FACTOR	PROCESSING FACTOR
OUTER RACE	1.00000E-07	4.11320E-02	1.00000E-01	0.00000E+00	0.00000E+00	1.19700E+00	1.00000E+00	7.70000E-02
INNER RACE	1.00000E-07	4.11320E-02	1.00000E-01	0.00000E+00	0.00000E+00	1.19700E+00	1.00000E+00	7.70000E-02

INITIAL OPERATING CONDITIONS

Cyl Roller Bearing Geo Mod

ROOM TEMPERATURE	(K)	2.94000E+02
HOUSING TEMPERATURE	(K)	2.94000E+02
SHAFT TEMPERATURE	(K)	2.94000E+02
ROLLING ELEMENT TEMPERATURE	(K)	2.94000E+02
CAGE TEMPERATURE	(K)	2.94000E+02
QUASI-STATIC CONSTRAINTS		1 1 0 1 1
CONSTRAINING LOAD FRACTION		5.00000E-02

		OUTER RACE	INNER RACE
ANGULAR VELOCITY	(rpm)	0.00000E+00	7.00000E+04
TEMPERATURE	(K)	2.94000E+02	2.94000E+02
MISALIGNMENT-Y	(rad)	0.00000E+00	0.00000E+00
MISALIGNMENT-Z	(rad)	0.00000E+00	0.00000E+00
TRANSLATIONAL CONSTRAINTS		1 1 1	1 1 1
ROTATIONAL CONSTRAINTS		1 1 1	1 1 1

BEARING COOLANT TYPE: No Coolant
NODE AT BASE TEMPERATURE: Housing/Outer
VARY GEOMETRY WITH TEMPERATURE: Yes

		X-COMP	Y-COMP	Z-COMP
APPLIED LOAD VECTOR	(N)	0.00000E+00	0.00000E+00	1.00000E+03
RELATIVE DISPLACEMENT VECTOR	(m)	0.00000E+00	0.00000E+00	0.00000E+00
APPLIED MOMENT VECTOR	(N.m)		0.00000E+00	0.00000E+00
GRAVITY VECTOR	(m/s**2)	0.00000E+00	0.00000E+00	-9.81000E+00
CAGE GEOM CEN POSITION	(m)	0.00000E+00	0.00000E+00	5.00000E-05
CAGE ANGULAR POSITION	(deg)	0.00000E+00	0.00000E+00	0.00000E+00
CAGE WHIRL TO ANG VEL RATIO		1.00000E+00		
CAGE ANG VEL TO EPICYCLIC VEL RATIO		1.00000E+00		

SCALE FACTORS AND OUTPUT CONTROLS

......SCALE FACTORS.......			......STEP SIZE INFO......					
LENGTH	(m)	4.00000E-03	MINIMUM	1.00000E-04	NO OF STEPS	4000		
LOAD	(N)	1.00000E+03	MAXIMUM	5.00000E-01	DATA CONTROL	1	500	
TIME	(S)	7.15843E-05	INITIAL	5.00000E-02	AUTO PLOTS	1	9	0
						0	0	0
			ERROR LIMIT	1.00000E-04	INT METHOD	5		

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Cyl Roller Bearing Geo Mod
=====	==========	==========	==========	==========	====================================

1. Rolling Element Parameters

Re no	...Orbital Position (deg)	....Contact Angle.... (deg) Outer Race	Inner Race	....Contact Load..... (N) Outer Race	Inner Race	...Contact Stress.... (Pa) Outer Race	Inner Race	..Major Half Width... (m) Outer Race	Inner Race	..Minor Half Width... (m) Outer Race	Inner Race
1	0.000E+00	0.000E+00	0.000E+00	7.693E+02	5.240E+02	1.354E+09	1.414E+09	1.963E-03	1.777E-03	9.215E-05	6.641E-05
2	4.500E+01	0.000E+00	0.000E+00	5.819E+02	3.366E+02	1.219E+09	1.191E+09	1.824E-03	1.595E-03	8.329E-05	5.638E-05
3	9.000E+01	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.036E-05	0.000E+00
4	1.350E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.036E-05	0.000E+00
5	1.800E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.036E-05	0.000E+00
6	2.250E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.036E-05	0.000E+00
7	2.700E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.036E-05	0.000E+00
8	3.150E+02	0.000E+00	0.000E+00	5.819E+02	3.366E+02	1.219E+09	1.191E+09	1.824E-03	1.595E-03	8.329E-05	5.638E-05

RE	...Orbital Velocity (rpm)	........Angular Velocity........ Amplitude (rpm)	Theta (deg)	Phi (deg)	...RE Ang Position... Theta (deg)	Phi (deg)	......Spin/Roll...... Outer Race	Inner Race	....Contact Loss..... (W) Outer Race	Inner Race	..Time Ave Wear Rate (m**3/s)
1	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.315E-01	2.334E-01	2.856E-15
2	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.616E-01	6.976E-02	1.390E-15
3	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.225E-16	1.452E-02	0.000E+00	1.300E-16
4	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.225E-16	1.452E-02	0.000E+00	1.300E-16
5	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.225E-16	1.452E-02	0.000E+00	1.300E-16
6	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.225E-16	1.452E-02	0.000E+00	1.300E-16
7	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.225E-16	1.452E-02	0.000E+00	1.300E-16
8	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.616E-01	6.976E-02	1.390E-15

RE no	....Slip Velocity.... (m/s) Outer Race	Inner Race	.....Trac Coeff...... Outer Race	Inner Race	....Iso Lub Film..... (m) Outer Race	Inner Race	...Thermal Red Fac... Outer Race	Inner Race	......Drag (N)	..Chrn Mom (N.m)	..Net Loss (W) (Drag+Chur)
1	8.312E-02	9.075E-02	7.979E-02	8.315E-02							
2	6.742E-02	6.412E-02	7.133E-02	6.926E-02							
3	3.388E-02	0.000E+00	4.444E-02	0.000E+00							
4	3.388E-02	0.000E+00	4.444E-02	0.000E+00							
5	3.388E-02	0.000E+00	4.444E-02	0.000E+00							
6	3.388E-02	0.000E+00	4.444E-02	0.000E+00							
7	3.388E-02	0.000E+00	4.444E-02	0.000E+00							
8	6.742E-02	6.412E-02	7.133E-02	6.926E-02							

RE no	Rel Axial Pos (m)	.Contact Deflection.. (m) Outer Race	Inner Race	....Race Flexing..... (m) Outer Race	Inner Race	..Contact Temp Rise.. (K) Outer Race	Inner Race	.....Contact Temp.... (K) Outer Race	Inner Race	Bulk Temp (K)	Convec HTC (W/m**2/K)
1	0.000E+00	5.706E-06	4.313E-06	0.000E+00	0.000E+00	1.498E-02	1.253E-02	2.940E+02	2.940E+02	2.940E+02	0.000E+00
2	0.000E+00	4.656E-06	3.090E-06	0.000E+00	0.000E+00	7.766E-03	4.166E-03	2.940E+02	2.940E+02	2.940E+02	0.000E+00
3	0.000E+00	2.413E-06	-1.534E-07	0.000E+00	0.000E+00	8.495E-04	0.000E+00	2.940E+02	2.940E+02	2.940E+02	0.000E+00
4	0.000E+00	2.413E-06	-5.638E-06	0.000E+00	0.000E+00	8.495E-04	0.000E+00	2.940E+02	2.940E+02	2.940E+02	0.000E+00
5	0.000E+00	2.413E-06	-7.910E-06	0.000E+00	0.000E+00	8.495E-04	0.000E+00	2.940E+02	2.940E+02	2.940E+02	0.000E+00
6	0.000E+00	2.413E-06	-5.638E-06	0.000E+00	0.000E+00	8.495E-04	0.000E+00	2.940E+02	2.940E+02	2.940E+02	0.000E+00
7	0.000E+00	2.413E-06	-1.534E-07	0.000E+00	0.000E+00	8.495E-04	0.000E+00	2.940E+02	2.940E+02	2.940E+02	0.000E+00
8	0.000E+00	4.656E-06	3.090E-06	0.000E+00	0.000E+00	7.766E-03	4.166E-03	2.940E+02	2.940E+02	2.940E+02	0.000E+00

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Cyl Roller Bearing Geo Mod
=====	==========	==========	==========	==========	====================================

1a. Load Distribution Along Roller no. 1

	Outer Race Contact: Axial Pos (m)	Geo Int (m)	Half Width (m)	Nor Load (N/m)	Trac Load (N/m)	Inner Race Contact: Axial Pos (m)	Geo Int (m)	Half Width (m)	Nor Load (N/m)	Trac Load (N/m)
1	-1.895E-03	5.258E-07	2.891E-05	7.206E+01	5.749E+00	1.715E-03	4.307E-07	2.145E-05	1.443E+04	-1.200E+03
2	-1.639E-03	2.333E-06	6.619E-05	3.774E+02	2.350E+01	1.483E-03	1.912E-06	4.910E-05	7.560E+04	-4.822E+03
3	-1.305E-03	4.298E-06	9.296E-05	7.440E+02	2.375E+01	1.181E-03	3.521E-06	6.895E-05	1.490E+05	-4.005E+03
4	-1.050E-03	5.503E-06	1.067E-04	9.790E+02	5.134E+00	9.500E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
5	-8.451E-04	5.706E-06	1.088E-04	1.019E+03	0.000E+00	7.650E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
6	-3.337E-04	5.706E-06	1.088E-04	1.019E+03	0.000E+00	3.020E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
7	3.337E-04	5.706E-06	1.088E-04	1.019E+03	0.000E+00	-3.020E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
8	8.451E-04	5.706E-06	1.088E-04	1.019E+03	0.000E+00	-7.650E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
9	1.050E-03	5.503E-06	1.067E-04	9.790E+02	5.134E+00	-9.500E-04	4.313E-06	7.718E-05	1.867E+05	0.000E+00
10	1.305E-03	4.298E-06	9.296E-05	7.440E+02	2.375E+01	-1.181E-03	3.521E-06	6.895E-05	1.490E+05	-4.005E+03
11	1.639E-03	2.333E-06	6.619E-05	3.774E+02	2.350E+01	-1.483E-03	1.912E-06	4.910E-05	7.560E+04	-4.822E+03
12	1.895E-03	5.258E-07	2.891E-05	7.206E+01	5.749E+00	-1.715E-03	4.307E-07	2.145E-05	1.443E+04	-1.200E+03

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Cyl Roller Bearing Geo Mod

2. Race and Cage Parameters

RE NO	GS NO	C C	RE/Cage Interaction: Geo Int (m)	Contact Force (N)	Con Ang THETA (deg)	Con Ang PHI (deg)	Contact Pos X (m)	Contact Pos Z (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	1	0	5.778E-05	0.000E+00	0.000E+00	0.000E+00	-6.405E-20	2.157E-04	0.000E+00	0.000E+00
1	2	0	5.778E-05	0.000E+00	0.000E+00	0.000E+00	-6.405E-20	-2.157E-04	0.000E+00	0.000E+00
2	1	0	9.314E-05	0.000E+00	0.000E+00	0.000E+00	-3.973E-20	2.293E-04	0.000E+00	0.000E+00
2	2	0	2.243E-05	0.000E+00	0.000E+00	0.000E+00	-1.031E-19	-2.293E-04	0.000E+00	0.000E+00
3	1	0	1.078E-04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.624E-04	0.000E+00	0.000E+00
3	2	1	-3.000E-35	1.516E+01	0.000E+00	0.000E+00	-7.604E-21	-2.624E-04	6.673E+01	3.402E-10
4	1	0	9.314E-05	0.000E+00	0.000E+00	0.000E+00	-3.973E-20	2.978E-04	0.000E+00	0.000E+00
4	2	0	2.243E-05	0.000E+00	0.000E+00	0.000E+00	2.682E-19	-2.978E-04	0.000E+00	0.000E+00
5	1	0	5.778E-05	0.000E+00	0.000E+00	0.000E+00	-6.405E-20	3.124E-04	0.000E+00	0.000E+00
5	2	0	5.778E-05	0.000E+00	0.000E+00	0.000E+00	-6.405E-20	-3.124E-04	0.000E+00	0.000E+00
6	1	0	2.243E-05	0.000E+00	0.000E+00	0.000E+00	2.682E-19	2.978E-04	0.000E+00	0.000E+00
6	2	0	9.314E-05	0.000E+00	0.000E+00	0.000E+00	-3.973E-20	-2.978E-04	0.000E+00	0.000E+00
7	1	0	7.781E-06	0.000E+00	0.000E+00	0.000E+00	-1.783E-19	2.624E-04	0.000E+00	0.000E+00
7	2	0	1.078E-04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-2.624E-04	0.000E+00	0.000E+00
8	1	0	2.243E-05	0.000E+00	0.000E+00	0.000E+00	2.682E-19	2.293E-04	0.000E+00	0.000E+00
8	2	0	9.314E-05	0.000E+00	0.000E+00	0.000E+00	-3.973E-20	-2.293E-04	0.000E+00	0.000E+00

LD NO	C C	Race/Cage Interaction: Geo Int (m)	Force (N)	Con Angle (deg)	Att Angle (deg)	Effective Dia Play (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	1	0.000E+00	2.052E-01	0.000E+00	0.000E+00	1.000E-04	6.903E-01	3.519E-12
2	1	0.000E+00	2.052E-01	0.000E+00	0.000E+00	1.000E-04	6.903E-01	3.519E-12

	Mass Center Position: Axial (m)	Radial (m)	Orbital (deg)	Orbital Velocity (rpm)	Angular Velocity: Amplitude (rpm)	Theta (deg)	Phi (deg)	Ang Position: Theta (deg)	Phi (deg)	Hoop Stress (Pa)	Time Ave Wear Rate (m**3/s)
Cage	0.000E+00	5.000E-05	0.000E+00	2.849E+04	2.849E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.473E-10
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.407E-14
IRace	0.000E+00	7.758E-06	0.000E+00	0.000E+00	7.000E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.532E+08	6.540E-15

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
0	0.000E+00	0.000E+00	0.000E+00	0.000E+00	Cyl Roller Bearing Geo Mod

3. Applied Parameters

	Forces in Inertial Frame: Comp-x (N)	Comp-y (N)	Comp-z (N)	Moments in Body-Fixed Frame: Comp-x (N.m)	Comp-y (N.m)	Comp-z (N.m)
Cage	0.000E+00	-7.374E-01	-1.557E+01	3.226E-01	1.847E-03	9.234E-05
Int ORace	0.000E+00	-1.376E+01	1.000E+03	5.640E-01	-1.847E-03	-9.234E-05
Int IRace	0.000E+00	8.763E+00	-1.000E+03	-1.742E-01	3.427E-17	1.745E-18
Ext ORace	0.000E+00	1.376E+01	-1.000E+03	-5.640E-01	1.847E-03	9.234E-05
Ext IRace	0.000E+00	-8.763E+00	1.000E+03	1.742E-01	-3.427E-17	-1.745E-18

Basic Fatigue Life	(hours)	1.369E+02
Modified Fatigue Life	(hours)	1.369E+02
Outer Race Fit	(m)	0.000E+00
Inner Race Fit	(m)	5.743E-06
Internal Clearance	(m)	1.774E-05
Total Power Loss	(W)	6.921E+01
Churning Loss Fraction		0.000E+00
Exit Fluid Temperature	(K)	0.000E+00

	....MC Acc in Inertial Frame.... (m/s**2)			...Ang Acc in Body-Fixed Frame... (rpm/s)			Bulk Temp (K)
	Comp-x	Comp-y	Comp-z	Comp-x	Comp-y	Comp-z	
Cage	0.000E+00	-4.660E+01	-9.936E+02	1.799E+02	1.987E+00	9.937E-02	2.940E+02
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.940E+02
IRace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.940E+02

Hsng/Bulk Temp (K)	2.940E+02
Hsng/Outer Temp (K)	2.940E+02
Hsng/Inner Temp (K)	2.940E+02
Shft/Bulk Temp (K)	2.940E+02
Shft/Outer Temp (K)	2.940E+02
Shft/Inner Temp (K)	2.940E+02

4. Time Step Summary

Step no	Time (s)	Outer Race Rotation (rev)	Inner Race Rotation (rev)	....Time Average Parameters.... Fatigue Life (hours)	Power Loss (W)	RE Orbital Vel Ratio	Cage Omega Ratio	Cage Whirl Ratio	Cage Wear Rate (m**3/s)	Average RE Excursion (m)
0	0.000E+00	0.000E+00	0.000E+00	1.369E+02	6.921E+01	4.070E-01	4.070E-01	4.070E-01	3.473E-10	1.269E-06

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
4000	7.618E+01	5.453E-03	0.000E+00	6.362E+00	Cyl Roller Bearing Geo Mod

1. Rolling Element Parameters

Re no	...Orbital Position (deg)	....Contact Angle.... (deg) Outer Race	Inner Race	....Contact Load..... (N) Outer Race	Inner Race	...Contact Stress.... (Pa) Outer Race	Inner Race	..Major Half Width... (m) Outer Race	Inner Race	..Minor Half Width... (m) Outer Race	Inner Race
1	5.722E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.709E+08	0.000E+00	1.485E-03	0.000E+00	6.037E-05	0.000E+00
2	6.172E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.037E-05	0.000E+00
3	6.621E+02	0.000E+00	0.000E+00	4.815E+02	2.362E+02	1.134E+09	1.035E+09	1.740E-03	1.474E-03	7.771E-05	4.928E-05
4	7.071E+02	0.000E+00	0.000E+00	7.575E+02	5.122E+02	1.346E+09	1.401E+09	1.955E-03	1.767E-03	9.164E-05	6.586E-05
5	7.522E+02	0.000E+00	0.000E+00	6.714E+02	4.262E+02	1.287E+09	1.306E+09	1.893E-03	1.688E-03	8.774E-05	6.155E-05
6	7.972E+02	0.000E+00	0.000E+00	3.231E+02	7.778E+01	9.713E+08	6.469E+08	1.580E-03	1.215E-03	6.701E-05	3.150E-05
7	8.422E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.708E+08	0.000E+00	1.485E-03	0.000E+00	6.037E-05	0.000E+00
8	8.872E+02	0.000E+00	0.000E+00	2.453E+02	0.000E+00	8.709E+08	0.000E+00	1.485E-03	0.000E+00	6.037E-05	0.000E+00

RE	...Orbital Velocity (rpm)	........Angular Velocity........ Amplitude (rpm)	Theta (deg)	Phi (deg)	...RE Ang Position... Theta (deg)	Phi (deg)	......Spin/Roll...... Outer Race	Inner Race	....Contact Loss..... (W) Outer Race	Inner Race	..Time Ave Wear Rate (m**3/s)
1	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.331E-02	0.000E+00	2.311E-17
2	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.331E-02	0.000E+00	2.009E-17
3	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	8.355E-02	2.412E-02	1.966E-17
4	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.700E-01	1.913E-01	2.206E-17
5	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.996E-01	1.173E-01	2.551E-17
6	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.793E-02	6.026E-04	2.592E-17
7	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.331E-02	0.000E+00	2.537E-17
8	2.849E+04	1.817E+05	1.800E+02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.331E-02	0.000E+00	2.560E-17

RE no	....Slip Velocity.... (m/s) Outer Race	Inner Race	.....Trac Coeff...... Outer Race	Inner Race	....Iso Lub Film..... (m) Outer Race	Inner Race	...Thermal Red Fac... Outer Race	Inner Race	......Drag (N)	..Chrn Mom (N.m)	..Net Loss (W) (Drag+Chur)
1	3.200E-02	0.000E+00	4.250E-02	0.000E+00							
2	3.200E-02	0.000E+00	4.249E-02	0.000E+00							
3	5.357E-02	4.537E-02	6.187E-02	5.523E-02							
4	7.469E-02	8.217E-02	7.552E-02	7.934E-02							
5	6.854E-02	7.149E-02	7.201E-02	7.374E-02							
6	3.964E-02	1.708E-02	5.008E-02	2.504E-02							
7	3.202E-02	0.000E+00	4.251E-02	0.000E+00							
8	3.201E-02	0.000E+00	4.251E-02	0.000E+00							

RE no	Rel Axial Pos (m)	.Contact Deflection.. (m) Outer Race	Inner Race	....Race Flexing..... (m) Outer Race	Inner Race	..Contact Temp Rise.. (K) Outer Race	Inner Race	.....Contact Temp.... (K) Outer Race	Inner Race	Bulk Temp (K)	Convec HTC (W/m**2/K)
1	0.000E+00	2.413E-06	-6.877E-06	0.000E+00	0.000E+00	7.676E-04	1.262E-07	2.940E+02	2.940E+02	2.940E+02	4.178E+05
2	0.000E+00	2.413E-06	-1.956E-06	0.000E+00	0.000E+00	7.674E-04	4.509E-07	2.940E+02	2.940E+02	2.940E+02	4.178E+05
3	0.000E+00	4.052E-06	2.345E-06	0.000E+00	0.000E+00	4.129E-03	1.535E-03	2.940E+02	2.940E+02	2.940E+02	4.178E+05
4	0.000E+00	5.642E-06	4.242E-06	0.000E+00	0.000E+00	1.204E-02	1.018E-02	2.940E+02	2.939E+02	2.939E+02	4.179E+05
5	0.000E+00	5.167E-06	3.699E-06	0.000E+00	0.000E+00	9.149E-03	6.529E-03	2.940E+02	2.938E+02	2.936E+02	4.184E+05
6	0.000E+00	2.994E-06	9.531E-07	0.000E+00	0.000E+00	1.521E-03	5.643E-05	2.940E+02	2.938E+02	2.936E+02	4.184E+05
7	0.000E+00	2.413E-06	-4.404E-06	0.000E+00	0.000E+00	7.678E-04	9.446E-08	2.940E+02	2.939E+02	2.939E+02	4.179E+05
8	0.000E+00	2.413E-06	-7.890E-06	0.000E+00	0.000E+00	7.677E-04	3.206E-07	2.940E+02	2.940E+02	2.940E+02	4.178E+05

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
4000	7.618E+01	5.453E-03	0.000E+00	6.362E+00	Cyl Roller Bearing Geo Mod

1a. Load Distribution Along Roller no. 1

	Outer Race Contact: Axial Pos (m)	Geo Int (m)	Half Width (m)	Nor Load (N/m)	Trac Load (N/m)	Inner Race Contact: Axial Pos (m)	Geo Int (m)	Half Width (m)	Nor Load (N/m)	Trac Load (N/m)
1	-1.434E-03	3.011E-07	2.122E-05	3.879E+01	1.648E+00	9.664E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
2	-1.240E-03	1.336E-06	4.858E-05	2.031E+02	4.635E+00	8.360E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
3	-9.878E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	6.658E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
4	-7.943E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	5.353E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
5	-6.396E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	4.311E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
6	-2.525E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	1.702E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
7	2.525E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	-1.702E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
8	6.396E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	-4.311E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
9	7.943E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	-5.353E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
10	9.878E-04	2.413E-06	6.747E-05	3.918E+02	-1.203E+00	-6.658E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
11	1.240E-03	1.336E-06	4.858E-05	2.031E+02	4.635E+00	-8.360E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01
12	1.434E-03	3.011E-07	2.122E-05	3.879E+01	1.648E+00	-9.664E-04	4.703E-09	1.745E-06	9.541E+01	-1.898E-01

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
4000	7.618E+01	5.453E-03	0.000E+00	6.362E+00	Cyl Roller Bearing Geo Mod

2. Race and Cage Parameters

RE NO	GS NO	C C	RE/Cage Interaction: Geo Int (m)	Contact Force (N)	Con Ang THETA (deg)	Con Ang PHI (deg)	Contact Pos X (m)	Contact Pos Z (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	1	0	3.895E-05	0.000E+00	0.000E+00	0.000E+00	1.580E-05	2.317E-04	0.000E+00	0.000E+00
1	2	0	7.661E-05	0.000E+00	0.000E+00	0.000E+00	-1.605E-05	-2.317E-04	0.000E+00	0.000E+00
2	1	0	7.379E-05	0.000E+00	0.000E+00	0.000E+00	-1.237E-05	2.171E-04	0.000E+00	0.000E+00
2	2	0	4.177E-05	0.000E+00	0.000E+00	0.000E+00	1.840E-06	-2.171E-04	0.000E+00	0.000E+00
3	1	0	9.154E-05	0.000E+00	0.000E+00	0.000E+00	-2.018E-05	2.307E-04	0.000E+00	0.000E+00
3	2	0	2.402E-05	0.000E+00	0.000E+00	0.000E+00	4.973E-05	-2.307E-04	0.000E+00	1.731E-15
4	1	0	9.051E-05	0.000E+00	0.000E+00	0.000E+00	-1.731E-05	2.638E-04	0.000E+00	0.000E+00
4	2	0	2.506E-05	0.000E+00	0.000E+00	0.000E+00	5.965E-05	-2.638E-04	0.000E+00	2.485E-14
5	1	0	9.860E-05	0.000E+00	0.000E+00	0.000E+00	-3.366E-06	2.959E-04	0.000E+00	0.000E+00
5	2	0	1.697E-05	0.000E+00	0.000E+00	0.000E+00	5.161E-05	-2.959E-04	0.000E+00	0.000E+00
6	1	0	6.696E-05	0.000E+00	0.000E+00	0.000E+00	1.302E-05	3.083E-04	0.000E+00	0.000E+00
6	2	0	4.861E-05	0.000E+00	0.000E+00	0.000E+00	-4.076E-07	-3.083E-04	0.000E+00	0.000E+00
7	1	0	3.622E-05	0.000E+00	0.000E+00	0.000E+00	4.237E-05	2.958E-04	0.000E+00	7.018E-15
7	2	0	7.934E-05	0.000E+00	0.000E+00	0.000E+00	-1.238E-05	-2.958E-04	0.000E+00	0.000E+00
8	1	0	2.346E-05	0.000E+00	0.000E+00	0.000E+00	6.466E-05	2.643E-04	0.000E+00	0.000E+00
8	2	0	9.210E-05	0.000E+00	0.000E+00	0.000E+00	-1.738E-05	-2.643E-04	0.000E+00	0.000E+00

LD NO	C C	Race/Cage Interaction: Geo Int (m)	Force (N)	Con Angle (deg)	Att Angle (deg)	Effective Dia Play (m)	Contact Loss (W)	Time Ave Wear Rate (m**3/s)
1	0	2.827E-06	0.000E+00	4.798E+01	0.000E+00	9.359E-05	0.000E+00	1.165E-13
2	0	2.827E-06	0.000E+00	4.798E+01	0.000E+00	9.359E-05	0.000E+00	1.165E-13

	Mass Center Position: Axial (m)	Radial (m)	Orbital (deg)	Orbital Velocity (rpm)	Angular Velocity: Amplitude (rpm)	Theta (deg)	Phi (deg)	Ang Position: Theta (deg)	Phi (deg)	Hoop Stress (Pa)	Time Ave Wear Rate (m**3/s)
Cage	-5.457E-08	4.536E-05	2.595E+02	3.140E+04	2.849E+04	1.137E-02	3.492E+02	1.846E-02	3.093E+01	3.354E+07	2.666E-13
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-1.687E+04	4.224E-16
IRace	0.000E+00	7.879E-06	0.000E+00	0.000E+00	7.000E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.532E+08	1.918E-16

step no	tau	Time (s)	Outer Race Rot (rev)	Inner Race Rot (rev)	(ADORE 5.0)
4000	7.618E+01	5.453E-03	0.000E+00	6.362E+00	Cyl Roller Bearing Geo Mod

3. Applied Parameters

	Forces in Inertial Frame: Comp-x (N)	Comp-y (N)	Comp-z (N)	Moments in Body-Fixed Frame: Comp-x (N.m)	Comp-y (N.m)	Comp-z (N.m)
Cage	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
Int ORace	0.000E+00	1.198E+01	1.003E+03	-3.887E-05	4.327E-18	2.143E-17
Int IRace	0.000E+00	-1.159E+01	-1.003E+03	-1.503E-04	-5.919E-18	-1.720E-17
Ext ORace	0.000E+00	-1.198E+01	-1.003E+03	3.887E-05	-4.327E-18	-2.143E-17
Ext IRace	0.000E+00	1.159E+01	1.003E+03	1.503E-04	5.919E-18	1.720E-17

Basic Fatigue Life	(hours)	1.347E+02
Modified Fatigue Life	(hours)	1.347E+02
Outer Race Fit	(m)	6.272E-09
Inner Race Fit	(m)	5.732E-06
Internal Clearance	(m)	1.779E-05
Total Power Loss	(W)	9.677E-01
Churning Loss Fraction		0.000E+00
Exit Fluid Temperature	(K)	2.939E+02

	....MC Acc in Inertial Frame.... (m/s**2) Comp-x	Comp-y	Comp-z	...Ang Acc in Body-Fixed Frame... (rpm/s) Comp-x	Comp-y	Comp-z	Bulk Temp (K)
Cage	0.000E+00	0.000E+00	-9.810E+00	0.000E+00	-6.448E+00	1.225E+00	2.939E+02
ORace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.940E+02
IRace	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.939E+02

Hsng/Bulk Temp (K)	2.940E+02
Hsng/Outer Temp (K)	2.940E+02
Hsng/Inner Temp (K)	2.940E+02
Shft/Bulk Temp (K)	2.939E+02
Shft/Outer Temp (K)	2.939E+02
Shft/Inner Temp (K)	2.939E+02

4. Time Step Summary

Step no	Time (s)	Outer Race Rotation (rev)	Inner Race Rotation (rev)	.....Time Average Parameters..... Fatigue Life (hours)	Power Loss (W)	RE Orbital Vel Ratio	Cage Omega Ratio	Cage Whirl Ratio	Cage Wear Rate (m**3/s)	Average RE Excursion (m)
4000	5.453E-03	0.000E+00	6.362E+00	2.784E+00	7.198E-02	8.345E-03	8.345E-03	8.770E-03	2.666E-13	2.472E-06

Normal termination: Last Step Number = 4000 Final Time = 7.6180E+01

Statistics of this Run

```
Minimum Step Size      =  4.06716E-03
Maximum Step Size      =  4.37787E-02
Last Step Size         =  2.65720E-02
Max Truncation         =  9.93185E-05
Total Derivative Calls =        43010
```